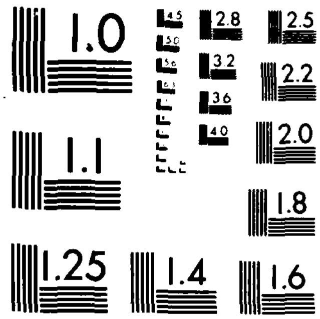
AD-A173 849 A FLIGHT DYNAMIC SIMULATION PROGRAM IN AIR-PATH AXES USING ACSL (ADVANCED..(U) AERONAUTICAL RESEARCH LABS MELBOURNE (AUSTRALIA) P W GIBBENS JUN 86 1/2

UNCLASSIFIED ARL-AERO-TM-380 F/G 12/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

12

ARL-AERO-TM-380 AR-004-472

AUSTRALIA

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORIES

MELBOURNE, VICTORIA

Aerodynamics Technical Memorandum 380

A FLIGHT DYNAMIC SIMULATION PROGRAM IN AIR-PATH AXES USING A.C.S.L.

by

P.W. GIBBENS

APPROVED FOR PUBLIC RELEASE.

JUNE 1986

AR-004-472

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
AERONAUTICAL RESEARCH LABORATORIES

Aerodynamics Technical Memorandum 380

A FLIGHT DYNAMIC SIMULATION PROGRAM IN AIR-PATH AXES USING A.C.S.L.

by

P.W. GIBBENS

SUMMARY

The six degrees of freedom dynamic equations of motion have been programmed in the Advanced Continuous Simulation Language (ACSL) for use in aircraft simulations at ARL. Air-path axes were chosen for the integration of the force equations, and body axes for the integration of the moment equations. The use of quaternions for the determination of the direction cosines has been described.

POSTAL ADDRESS: Director, Aeronautical Research Laboratories,
P.O. Box 4331, Melbourne, Victoria, 3001, Australia.

CONTENTS

NOTATION

a_x, a_y, a_z	Linear accelerations in body axes (g's)
a_n	Normal acceleration (g's)
C_i (i=0,9)	Inertia constants

Non-dimensional aerodynamic coefficients

C_{D_e}	Equilibrium drag coefficient
C_{D_V}	Derivative of drag coefficient with respect to velocity
C_{D_α}	Derivative of drag coefficient with respect to angle of attack
C_{L_e}	Equilibrium lift coefficient
C_{L_q}	Derivative of lift coefficient with respect to pitch rate
C_{L_V}	Derivative of lift coefficient with respect to velocity
C_{L_α}	Derivative of lift coefficient with respect to angle of attack
$C_{L_{\dot{\alpha}}}$	Derivative of lift coefficient with respect to rate of change of angle of attack
C_{m_q}	Derivative of moment coefficient with respect to pitch rate
C_{m_V}	Derivative of moment coefficient with respect to velocity
C_{m_α}	Derivative of moment coefficient with respect to angle of attack
$C_{m_{\dot{\alpha}}}$	Derivative of moment coefficient with respect to rate of change of angle of attack
C_{T_e}	Equilibrium thrust coefficient
C_{T_V}	Derivative of thrust coefficient with respect to velocity

- - - - - - - - - - - - - - - - - - -

NOTATION (Contd.)

F_{GX}, F_{GY}, F_{GZ}	Gravitational forces	(N)
F_{XB}, F_{YB}, F_{ZB}	Total forces in body axes	(N)
F_{XS}, F_{YS}, F_{ZS}	Total forces in stability axes	(N)
F_{XW}, F_{YW}, F_{ZW}	Total forces in air-path axes	(N)
g	Gravitational constant	(ms^{-2})
I_{XX}, I_{YY}, I_{ZZ}, I_{XZ}	Moments and product of inertia	(Kg m^2)
L, M, N	Total applied moments in body axes	(Nm)
L_A, M_A, N_A	Applied aerodynamic moments in stability axes	(Nm)
L_P, M_P, N_P	Applied propulsive moments in body axes	(Nm)
L_i, M_i, N_i (i=1,3)	Direction cosines	
m	Aircraft mass	(Kg)
P, Q, R	Angular velocity components about body axes	(rad. s^{-1})
P_S, R_S	Pitch and yaw rates about stability axes	(rad.s^{-1})
RANGE	Aircraft range	(m)
U_B, V_B, W_B	Body axes velocity components	(ms^{-1})
V_E, V_{EK}	Equivalent airspeed	(ms^{-1}, Knots)
V_{GR}	Aircraft ground speed	(ms^{-1})
V_{NE}, V_{EE}, V_{DE}	Components of velocity relative to the earth	(ms^{-1})
V_T	True airspeed	(ms^{-1})
X_A, Y_A, Z_A	Applied aerodynamic forces in stability axes	(N)
X_P, Y_P, Z_P	Applied propulsive forces in body axes	(N)
X_B, Y_B, Z_B	Body axes reference frame	
X_E, Y_E, Z_E	Earth axes reference frame	
X_S, Y_S, Z_S	Stability axes reference frame	

NOTATION (Contd.)

X_W, Y_W, Z_W	Air-path axes reference frame	
α	Angle of attack	(rad)
β	Angle of sideslip	(rad)
γ	Flight path angle	(rad)
θ, ϕ, ψ	Euler angles of pitch, roll (bank) and yaw (heading)	(rad)
λ	Angle of climb	(rad)
τ_i (i=0,3)	Quaternion components	
χ	Angle of track, east of north	(rad)

Subscripts

A	Aerodynamic contribution
B	Body axes
E	Earth axes
P	Propulsive contribution
S	Stability axes
W	Air-path axes

A dot over a variable denotes the first derivative with respect to time.

1. INTRODUCTION

The aircraft behavioural Studies - Fixed Wing (ABS - FW) group of Aerodynamics Division at the Aeronautical Research Laboratories (ARL) is concerned with the flight dynamic behaviour of fixed wing aircraft, and has the responsibility for developing flight dynamic computer models of the advanced high performance aircraft operated by the RAAF.

This memorandum documents the basic six degrees of freedom dynamic equations of motion incorporated in the associated simulation program SDOFAP which was written using Advanced Continuous Simulation Language (ACSL). The program has been developed for general use by the ABS-FW group for aircraft simulation studies. The program presented in Ref. (1), which was originally written using Earth axes, has been modified to use Air-path axes for integration of the force equations to allow the linear analysis capabilities of ACSL to be utilised more conveniently.

Section 2 of the memorandum gives a description of the program and its structure, while section 3 deals with the axes systems, their selection and transformation, the use of quaternions in determining aircraft attitude, and the six degrees of freedom equations of motion. Section 4 contains an example application of the program which demonstrates the analytical capabilities of the ACSL language and illustrates various alternative presentations of the output data.

2. ACSL AIRCRAFT FLIGHT DYNAMIC SIMULATION PROGRAM - SDOFAP

The six degree of freedom aircraft simulation program SDOFAP has been written in air-path axes and utilises the analytical features of the ACSL language. A description of ACSL and its utilisation is given in Ref. (2).

The ACSL program structure contains three primary regions, each dealing with specific segments of program execution; these are the INITIAL, DYNAMIC and DERIVATIVE sections. Fig. 1 illustrates the program flow.

The INITIAL section serves the following purposes;

1. Aircraft configuration and aerodynamic data are read in at subroutine level from an input file. This file is prepared by the setup program FTCHOO.

2. The initialisation of state and control variables and specification of atmospheric data and other constants is performed.

3. The setting and updating of prescribed control variables and runtime parameters for particular simulation runs is performed.

Time history calculation is carried out within the DYNAMIC section. This portion of the program primarily administers the trimming procedures, prepares output variables and manages the generation and logging of time dependent results.

The DERIVATIVE section contains the details of the six degrees of freedom flight model.

Trimming of the aircraft equations is performed by a user supplied trimming subroutine. The routine POWIT used in the example application calls the subroutine EVAL, which gives access to the DERIVATIVE section. Because ACSL does not allow the use of common

statements, EVAL must be appended to the ACSL program and variable values made available by use of the ACSL inclusion character '$' as described in Ref. (2).

To provide the trim routine with initial state and control variables, an approximation is made before trimming. When trimming is complete, control is transferred back to the INITIAL section.

A description of the six degree of freedom equations is given in Section 3, and complete program listings are presented in Appendix 1.

3. SDOFAP SIMULATION MODEL

3.1 Definition of Axes Systems

All axes systems are assumed to be orthogonal, right-handed triads, and are shown in figure 2.

(i) Earth Axes (X_E, Y_E, Z_E)

The origin is at a point fixed on the earth's surface, typically at the runway threshold and on the centreline. The x-axis points North, the Y-axis East, and the Z-axis 'down' toward the centre of the earth. It is assumed that the earth is flat and non-rotating, such that the earth axes are regarded as an inertial frame.

(ii) Body Axes (X_B, Y_B, Z_B)

Body axes are fixed on the aircraft with the origin located at the aircraft centre of gravity. The aircraft is assumed to be rigid, with the X-axis parallel to the horizontal fuselage reference line and pointing 'forward', the Y-axis pointing to starboard (right), and the Z-axis 'downward'.

(iii) Stability Axes (X_X, Y_S, Z_S)

Stability axes are a special set of body axes used primarily in the study of small disturbances from a steady reference flight condition. Aerodynamic data are frequently presented in stability axes. These axes are displaced from the body frame by the angle of attack, α, such that the X-axis in the steady-state is aligned with the projection of the relative wind vector on the aircraft plane of symmetry; the Y-axis points to starboard, and the Z-axis 'downward'.

(iv) Air-Path Axes (X_W, Y_W, Z_W)

Air-path axes differ from body axes by the angle of attack, α, and the angle of sideslip, β. The transformation from body to air-path axes is accomplished as shown in figure 2, by first pitching through $-\alpha$, to coincide with the stability axes, and then yawing through β. The origin is located at the aircraft centre of gravity and the X-axis is aligned with the relative wind vector.

Note: When the wind velocity components are zero, the air-path axes coincide with the flight-path axes as defined by Fogarty and Howe (4). Etkin (3) refers to the air-path axes as the air-trajectory reference frame (or wind axes).

3.2 Selection of Axes

The various options for selection of axes systems are discussed in Ref. (3) and Ref. (4). With the development of high speed digital computers with large data storage, the selection of axes systems has become less critical.

(i) Force Equations.

In recent years, the use of body axes for the computation and integration of force equations has become less popular. Ref. (1), from which this program was developed, employs earth axes to suit the particular application in that report. However, air-path axes have been selected here because when used in conjunction with the ACSL linear analysis procedures they yield parameters which can be used directly for simulation validation and assessment.

(ii) Moment Equations

The body axes system is the natural choice for the solution of the rotational equations of motion because of the important advantage of constant moments of inertia when calculating the moments and angular motion of the aircraft.

3.3 Aircraft Attitude Determination

The attitude of an aircraft is defined in terms of the traditional Euler angles, ψ (heading angle), θ (pitch attitude), and ϕ (roll, or 'bank' angle). In order to avoid the problems associated with the singularity in the Euler 'rate' equations, which occurs when $\theta = \pm 90°$, quaternion components (5) or direction cosines may be used in the integration step.

Quaternion components were chosen for the following reasons:

(i) their time derivates are always finite and continuous, whereas those of the Euler angles possess singularities;

(ii) the computations remain accurate as θ approaches $90°$;

(iii) it is a four parameter method consisting of four integrations with one constraint equation, whereas direction cosines, in principle, require nine integrations and six constraint equations.

The quaternion components are expressible in terms of Euler angles as follows:

$$\tau_0 = \cos\phi/2 \cos\theta/2 \cos\psi/2 + \sin\phi/2 \sin\theta/2 \sin\psi/2$$

$$\tau_1 = \sin\phi/2 \cos\theta/2 \cos\psi/2 - \cos\phi/2 \sin\theta/2 \sin\psi/2$$

$$\tau_2 = \cos\phi/2 \sin\theta/2 \cos\psi/2 + \sin\phi/2 \cos\theta/2 \sin\psi/2$$

$$\tau_3 = \cos\phi/2 \cos\theta/2 \sin\psi/2 - \sin\phi/2 \sin\theta/2 \cos\psi/2$$

The quaternion component time derivatives are given by,

$$\begin{aligned}
\dot{\tau}_0 &= -\ 1/2\ (P\tau_1 + Q\tau_2 + R\tau_3) \\
\dot{\tau}_1 &= \quad 1/2\ (P\tau_0 - Q\tau_3 + R\tau_2) \\
\dot{\tau}_2 &= \quad 1/2\ (P\tau_3 + Q\tau_0 - R\tau_1) \\
\dot{\tau}_3 &= -\ 1/2\ (P\tau_2 - Q\tau_1 - R\tau_0)
\end{aligned} \qquad (2)$$

where P,Q,R are angular velocity components about body axes, and

$$\tau_0^2 + \tau_1^2 + \tau_2^2 + \tau_3^2 = 1 \qquad (3)$$

Failure to normalize the quaternion components at each iteration can result in the integration becoming unstable.

Euler angles may be derived from the quaternion components or from direction cosines by using the following relationships,

$$\phi = \tan^{-1}\left[\frac{\tau_2\tau_3 + \tau_0\tau_1}{\tau_0^2 + \tau_3^2 - 1/2}\right] = \tan^{-1}\left[\frac{M_3}{N_3}\right] \qquad (4)$$

$$\theta = \tan^{-1}\left[\frac{\tau_0\tau_2 - \tau_1\tau_3}{[(\tau_0^2+\tau_1^2 - 1/2)^2 + (\tau_1\tau_2 + t_0\tau_3)^2]^{1/2}}\right] = \tan^{-1}\left[\frac{-L_3}{(L_1^2+L_2^2)^{1/2}}\right] \qquad (5)$$

$$\psi = \tan^{-1}\left[\frac{\tau_1\tau_2 + \tau_0\tau_3}{\tau_0^2+\tau_1^2-1/2}\right] = \tan^{-1}\left[\frac{L_2}{L_1}\right] \qquad (6)$$

The initial Euler angles are used to determine the initial quaternion components, which are in turn used to calculate the direction cosine parameters for use in axes transformation computations. The quaternion components are updated at each iteration, using equation (2), such that the direction cosines are recalculated for use in the equations of motion, while the Euler angles are calculated as output data only.

3.4 Axes Transformation

Transformation of a set of variables from body axes to earth axes (or vice-versa) is conveniently achieved by use of direction cosines (7), which are obtained in terms of the quaternion components by the following relationships,

$$
\begin{aligned}
L_1 &= 2\,(\tau_0^{\,2} + \tau_1^{\,2}) - 1 \\
L_2 &= 2\,(\tau_1\tau_2 + \tau_0\tau_3) \\
L_3 &= 2\,(\tau_1\tau_3 - \tau_0\tau_2) \\
M_1 &= 2\,(\tau_1\tau_2 - \tau_0\tau_3) \\
M_2 &= 2\,(\tau_0^{\,2} + \tau_2^{\,2}) - 1 \\
M_3 &= 2\,(\tau_2\tau_3 + \tau_0\tau_1) \\
N_1 &= 2\,(\tau_1\tau_3 + \tau_0\tau_2) \\
N_2 &= 2\,(\tau_2\,\tau_3 - \tau_0\tau_1) \\
N_3 &= 2\,(\tau_0^{\,2} + \tau_3^{\,2}) - 1
\end{aligned}
\tag{7}
$$

3.5 Equations of Motion

Figure 3 is a summary of the overall six degrees of freedom dynamic equations for the case of a flat earth.

(i) Force Equations

The aerodynamic force components are frequently computed along stability axes, and the propulsive force components are usually supplied in body axes. Resolution along stability axes is given by:

$$
\begin{aligned}
F_{XS} &= X_P \cos\alpha + Z_P \sin\alpha + X_A + F_{GX} \\
F_{YS} &= Y_P + Y_A + F_{GY} \\
F_{ZS} &= Z_P \cos\alpha - X_P \sin\alpha + Z_A + F_{GZ}
\end{aligned}
\tag{8}
$$

where F_{GX}, F_{GY}, F_{GZ} are the gravitational forces resolved into stability axes through use of the direction cosines.

$$\begin{aligned} F_{GX} &= L_3 mg\cos\alpha + N_3 mg\sin\alpha \\ F_{GY} &= M_3 mg \\ F_{GZ} &= -L_3 mg\sin\alpha + N_3 mg\cos\alpha \end{aligned} \qquad (9)$$

'g' is assumed to be constant such that the calculated altitude in figure 3 is the geopotential height, as used in standard atmosphere calculations.

The total force components in air-path axes are obtained by transformation of the forces in stability axes through the sideslip angle β.

$$\begin{aligned} F_{XW} &= F_{XS}\cos\beta + F_{YS}\sin\beta \\ F_{YW} &= -F_{XS}\sin\beta + F_{YS}\cos\beta \\ F_{ZW} &= F_{ZS} \end{aligned} \qquad (10)$$

The state variable derivatives are obtained from the dynamic equations;

$$\begin{aligned} \dot{\alpha} &= (Q\cos\beta - P_S\sin\beta + F_{ZW}/mV_T)/\cos\beta \\ \dot{\beta} &= F_{YW}/mV_T - R_S \\ \dot{V}_T &= F_{XW}/m \end{aligned} \qquad (11)$$

where m is the aircraft mass, and virtual mass effects are ignored. P_S and R_S are the angular rates about the X and Z axes respectively, in stability axes.

$$P_S = P \cos\alpha + R \sin\alpha$$
$$R_S = - P \sin\alpha + R \cos\alpha \tag{12}$$

The velocity components of the aircraft relative to the air may be obtained in body axes, if required, from the state variables, after integration of equation (11).

$$U_B = V_T \cos\alpha \cos\beta$$
$$V_B = V_T \sin\beta \tag{13}$$
$$W_B = V_T \sin\alpha \cos\beta$$

Wind components are introduced to give the components of aircraft velocity relative to the earth.

$$V_{NE} = L_1 U_B + M_1 V_B + N_1 W_B - V_{WN}$$
$$V_{EE} = L_2 U_B + M_2 V_B + N_2 W_B - V_{WE} \tag{14}$$
$$V_{DE} = L_3 U_B + M_3 V_B + N_3 W_B - V_{WD}$$

where V_{WN}, V_{WE}, V_{WD} are the wind components North, East and Down respectively with respect to the earth.

Flight path parameters are derived directly from the earth axes velocity vector components using equations (15) to (17).

Ground speed,

$$V_{GR} = (V_{NE}^2 + V_{EE}^2)^{\frac{1}{2}} \tag{15}$$

Climb angle,

$$\lambda = \mathrm{Tan}^{-1}\, [V_{DE}/V_{GR}] \tag{16}$$

Angle of Track,

$$\chi = \mathrm{Tan}^{-1}\, [V_{EE}/V_{NE}] \tag{17}$$

also, equation (18) gives the flight path angle α.

$$\gamma = \theta - \alpha \tag{18}$$

The positional coordinates of the aircrafts centre of gravity are deduced by integrating the earth axes velocity vector components:

$$\dot{X}_E = V_{NE} \tag{19}$$

$$\dot{Y}_E = V_{EE} \tag{20}$$

$$\dot{Z}_E = -V_{DE} \tag{21}$$

to give distance North, distance East and Altitude respectively, where altitude (ALT) = $- Z_E$.

The additional parameter, the Range of the aircraft C.G. from the runway threshold is calculated using the definition:

$$RANGE = (X_E^2 + Y_E^2)^{1/2} \tag{22}$$

The linear accelerations sensed by accelerometers mounted at the c.g. and aligned along the body axes are computed from the applied aerodynamic and propulsion forces as follows:

$$\begin{aligned} F_{XB} &= X_A \cos\alpha - Z_A \sin\alpha + X_P \\ F_{YB} &= Y_A + Y_P \\ F_{ZB} &= X_A \sin\alpha + Z_A \cos\alpha + Z_P \end{aligned} \tag{23}$$

The linear accelerations are then given by

$$\begin{aligned} a_x &= F_{XB}/mg \\ a_y &= F_{YB}/mg \\ a_z &= F_{ZB}/mg \end{aligned} \tag{24}$$

In aircraft operations reference is more commonly made to the normal acceleration or 'load factor' which is defined as:

$$a_n = -a_z \tag{25}$$

(ii) **Moment Equations**

The total moments acting on an aircraft consist of aerodynamic and powerplant components. The powerplant components which include

gyroscopic moments due to powerplant rotors, and thrust alignment moments, are normally given in body axes. The aerodynamic moments are, like the aerodynamic forces, frequently given in stability axes (3). If the aerodynamic moments are given in body axes, the simplification of equation (26) is obvious (i.e. set $\alpha=0$).

$$\begin{aligned} L &= L_A \cos\alpha - N_A \sin\alpha + L_P \\ M &= M_A + M_P \\ N &= L_A \sin\alpha + N_A \cos\alpha + N_P \end{aligned} \tag{26}$$

If it is assumed that the aircraft has a plane of symmetry, such that the products of inertia I_{YZ} and I_{XY} are zero, then the body axes angular accelerations can be calculated by using equation (27).

$$\begin{aligned} \dot{P} &= L.C_1 + N.C_2 + (P.C_3 + R.C_4)\,Q \\ \dot{Q} &= M.C_5 + (R^2 - P^2)\,C_6 + R.P.C_7 \\ \dot{R} &= N.C_8 + L.C_2 + (P.C_9 - R.C_3)\,Q \end{aligned} \tag{27}$$

where

$$\begin{aligned} C_0 &= I_{XX}I_{ZZ} - I_{XZ}^2 \\ C_1 &= I_{ZZ}/C_0 \\ C_2 &= I_{XZ}/C_0 \\ C_3 &= C_2\,(I_{XX} - I_{YY} + I_{ZZ}) \\ C_4 &= C_1\,(I_{YY} - I_{ZZ}) - C_2 I_{XZ} \\ C_5 &= 1/I_{YY} \\ C_6 &= C_5 I_{XZ} \\ C_7 &= C_5\,(I_{ZZ} - I_{XX}) \\ C_8 &= I_{XX}/C_0 \\ C_9 &= C_8\,(I_{XX} - I_{YY}) + C_2 I_{XZ} \end{aligned} \tag{28}$$

The constants C_0 to C_9 are evaluated during initialization.

The aircraft angular velocity components may then be obtained by integrating equation (27).

4. EXAMPLE APPLICATION: SIMULATION OF THE LONGITUDINAL MOTION OF A LIGHT AIRCRAFT

4.1 Background.

In Ref. (6), the effects of power on the longitudinal aerodynamic characteristics of a single engined, propeller driven aeroplane were investigated. The simulation developed in that report was written in FORTRAN 66 code for the DEC system 10 computer. The subroutines employed a model of propulsion effects on the aerodynamics of a competition aerobatic aircraft. These subroutines have been updated to FORTRAN 77 level and transferred to an ELXSI 6400 computer for use in the SDOFAP model.

Appendix 2 presents a listing of the simulation program using the master program SDOFAP.ACSL, plus the associated subroutines. Since the aerodynamic and propulsive forces are inter-related in the power effects modelling, they have been supplied to the ACSL program already combined in stability axes, through the subroutine AERO, and the subroutine PROP which normally supplies the propulsive forces was bypassed.

The calling sequence of the power effects subroutines is illustrated in Fig. 4 together with brief explanations of their individual purposes.

Preparation of the input data for this example is demonstrated in Appendix 3 together with a listing of the data preparation program FTCHOO. Some of the variables listed in the data tables relate to the original power effects simulation program of Ref. (6) and are not used in the ACSL program. It is recommended however that this format be employed for data entry.

4.2 Time Histories

Time histories of the longitudinal dynamic behaviour of the light aircraft have been generated in Appendix 4. Input data and command files have been included to demonstrate the use of ACSL run-time commands.

Both the short period and phugoid modes have been analysed to illustrate the various plotting capabilities of ACSL.

4.3 Eigen Analysis

The eigen analysis of Appendix 5 gives results for the same flight case as in section 4.2.

By eliminating the lateral variables using the FREEZE facility, results for the longitudinal modes only are obtained. The first eigenvalue and its accompanying eigenvector are associated with the quaternions, the second and third with the phugoid mode and the fourth and fifth with the short period mode.

4.4 Jacobian Analysis

The purpose of this extension to the program is to extract the non-dimensional aerodynamic derivatives from the elements of the non-dimensional Jacobian matrix.

The ANALYZ 'JACOB' run-time command causes ACSL to calculate a Jacobian matrix around the current trim point in state space. In order to gain access to this matrix it is necessary to call the subroutine *INTERM* from the ACSL library subroutine ZZEIGC. The resulting matrix comprises the linearised small disturbance equations of longitudinal motion given in Ref. (3) Eq. (5.13-19). The matrix contains ten non-zero coefficients in rows 1 to 3 from which eleven unknown longitudinal derivatives are to be determined. It is also noted that two elements in column 4 become zero for zero flight path angle and that the denominators in rows 2 and 3 contain the term $(2\mu + C_{L_{\dot{\alpha}}})$ in which 2μ is almost three orders of magnitude greater tnan $C_{L_{\dot{\alpha}}}$. The equations are therefore ill-conditioned and in order to obtain satisfactory estimates for the eleven unknowns, a number of assumptions are proposed.

Two options are included in the program. the first option can only be used if the flight-path angle γ is significantly greater than zero and includes the assumption $C_{T_V} = -3\ C_{T_e}$ for propellor driven aircraft and $C_{T_V} = -2\ C_{T_e}$ for jet or rocket powered aircraft. (Ref. (3) Section 7.8). The remaining ten derivatives are obtained from the ten matrix elements. Evaluation of this method has shown that the derivatives $C_{L_{\dot{\alpha}}}$ and C_{L_q} become inaccurate if $|\gamma| < 2.0$ degrees, and that the other derivatives are unreliable if $|\gamma| < 0.2$ degrees.

In the second option, it is further assumed that tne derivatives $C_{m_{\dot{\alpha}}}$ and $C_{L_{\dot{\alpha}}}$ are related to C_{m_q} and C_{L_q} directly through the downwasn rate $d\varepsilon/d\alpha$ (Ref. (3) Section 7.10) and that C_{m_q} and $C_{m_{\dot{\alpha}}}$ are related to C_{L_q} and $C_{L_{\dot{\alpha}}}$ via the tailplane non-dimensional moment arm. These additional constraints assume that the contributions to the q and $\dot{\alpha}$ derivatives are entirely due to the tailplane lift. Seven of the unknown

derivatives are derived from the matrix elements and the remainder from the above assumed relationships.

The second option provides more accurate estimates of the derivatives for cases where the assumed constraints are applicable.

5. CONCLUSIONS

The six degrees of freedom dynamic equations of aircraft motion have been programmed using the Advanced Continuous Simulation Language for use in aircraft simulations. The air-path axes system was chosen for the integration of the force equations, while the moment equations are integrated in body axes. Euler angles and direction cosines have been calculated by use of quaternion components.

Details of an example application have been provided to illustrate the use of the program and its potential. Time histories, eigen and jacobian analyses have been demonstrated.

REFERENCES

1. HALL, P.H. "A Set of Flight Dynamic Equations for Aircraft Simulation". ARL-AERO-TECH-MEMO-339. June 1982

2. MITCHELL & GAUTHIER, ASSOC., INC. "Advanced Continuous Simulation Language (ACSL). User Guide/Reference Manual".

3. ETKIN, B. "Dynamics of Atmospheric Flight". Wiley & Sons, 1972.

4. FOGARTY, L.E. & HOWE, R.M. "Computer Mechanization of Six Degree of Freedom Flight Equations." NASA CR-1344, May 1969.

5. ROBINSON, A.C. "On the Use of Quaternions in Simulation of Rigid-Body Motion." WADC TR 58-17, December 1958.

6. MARTIN, C.A. "Power effects on the longitudinal characteristics of a single-engine, propeller driven aircraft." ARL-AERO-REPORT-1[illegible]7, February 1983.

ACKNOWLEDGEMENTS

The author would like to thank Mr C.A. Martin for the expert guidance tendered throughout the course of the development of this program, and Mr R.H. Perrin for his generous help in the refinement of the routines described in Section 4.4. Appreciation is extended to all members of ABS-FW group for their enthusiastic support.

```
PROGRAM
    INITIAL
        }Statements performed before the run begins. State variables do not contain
        }the initial conditions yet.
    END
    DYNAMIC
        DERIVATIVE
            }Statements needed to calculate derivatives of each INTEG
            }statement. The dynamic model.
        END
        }Statements executed every communications interval.
    END
    TERMINAL
        }Statements executed when the termination condition TERMT becomes
        }true.
    END
END
```

Outline of Explicitly Structured Program

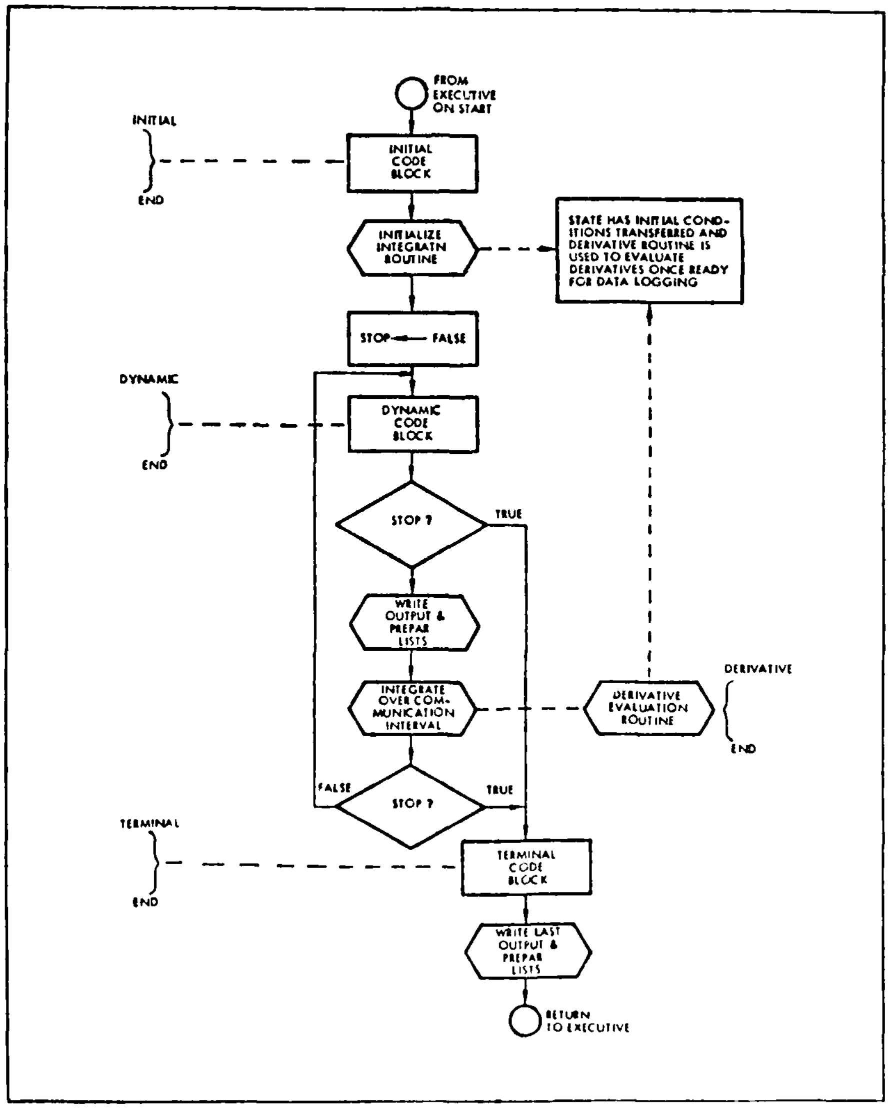


Figure 1 . Main Program Loop of ACSL Model

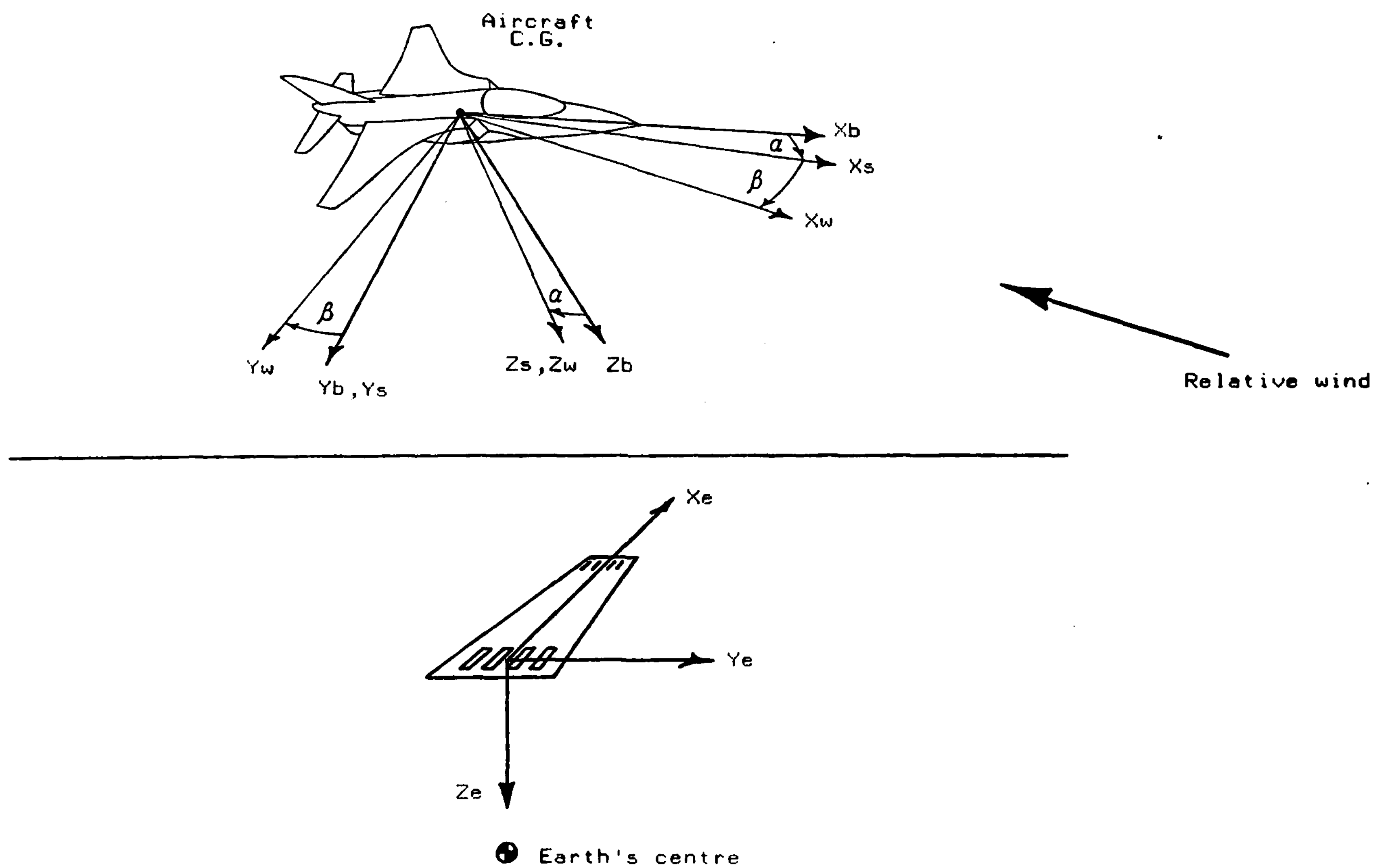


FIG 2: RELATIONSHIP BETWEEN EARTH AXES, BODY AXES, STABILITY AXES AND AIR-PATH AXES.

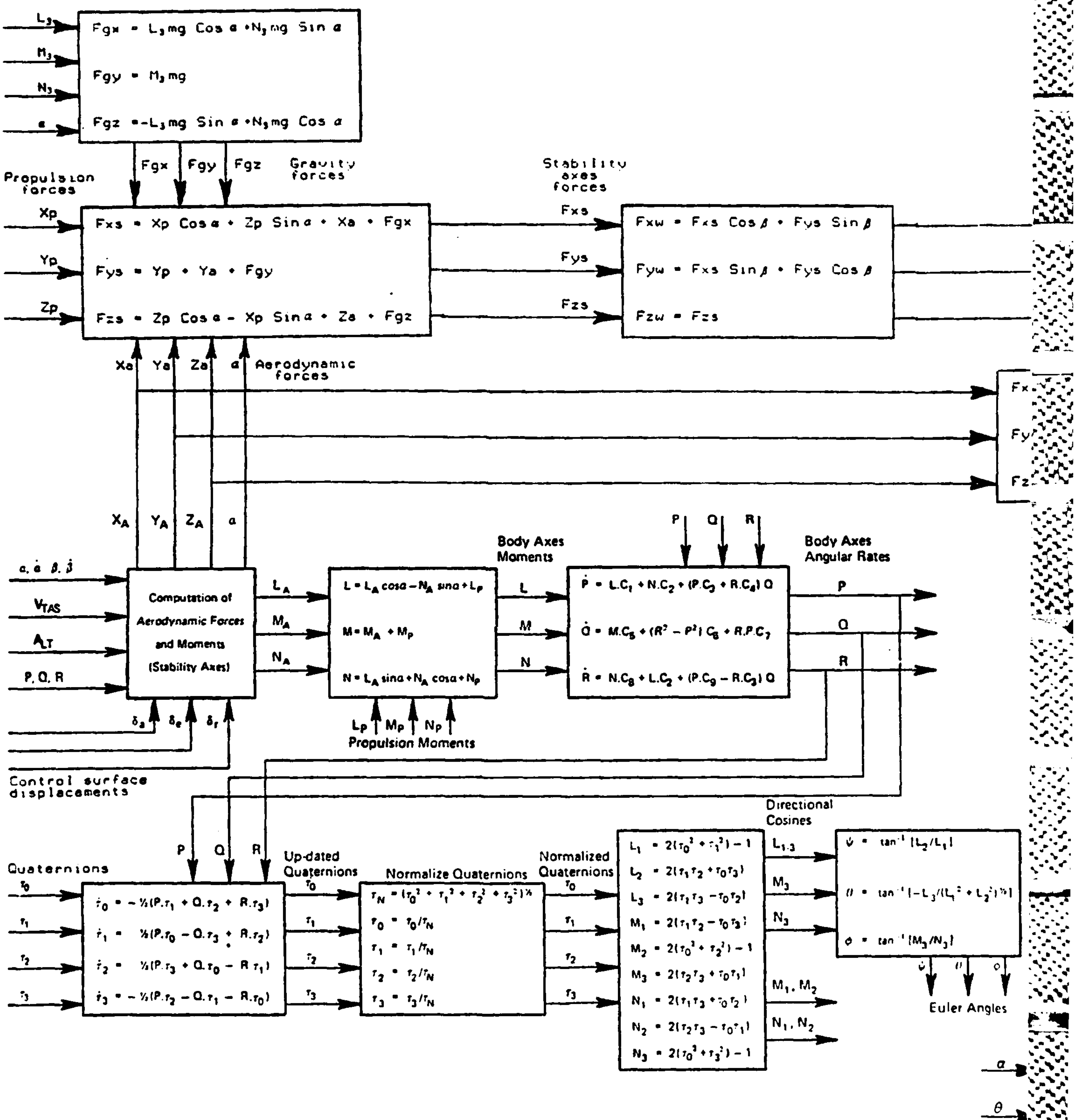
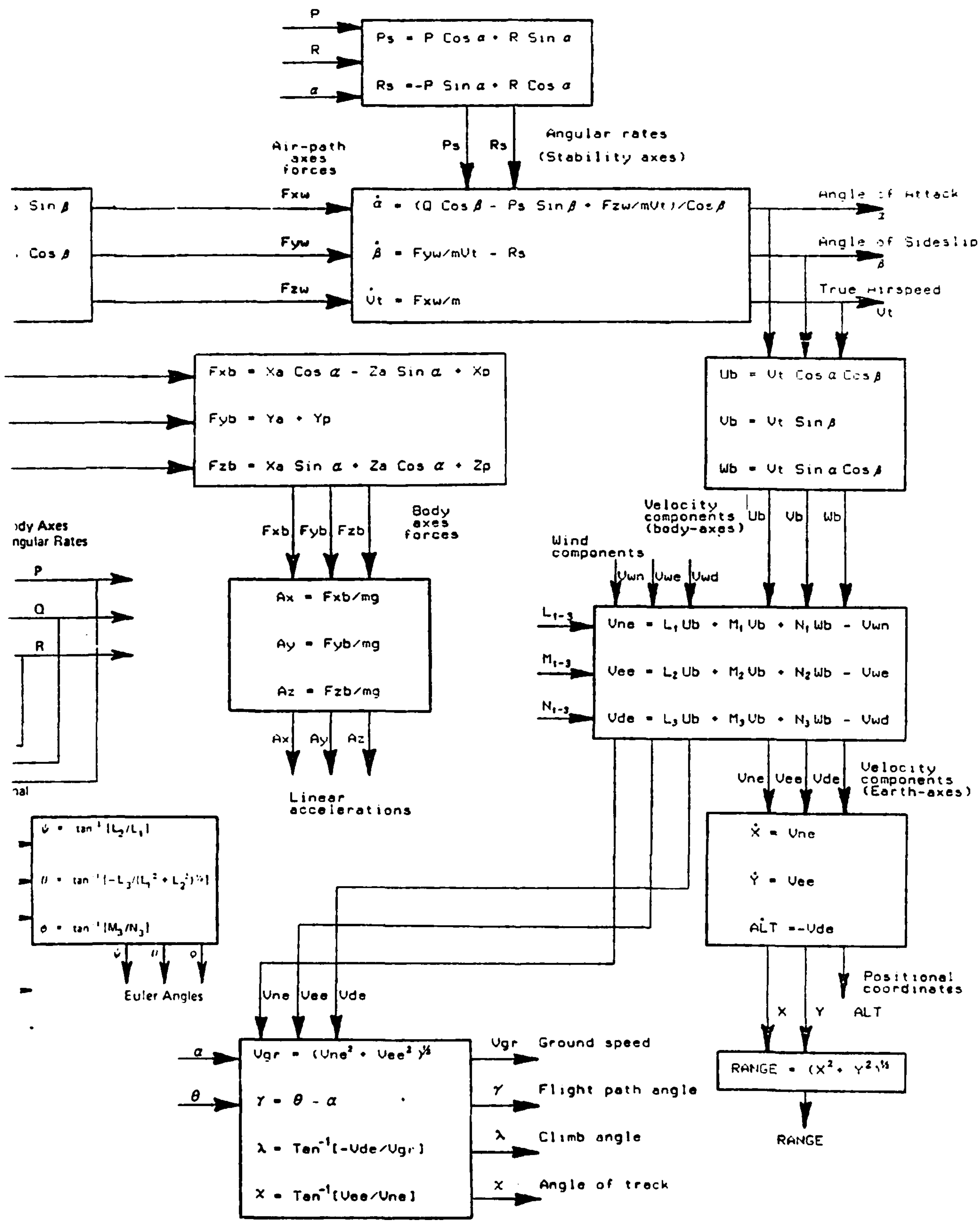


FIG 3: BLOCK DIAGRAM OF COMBINED AIR-PATH AXES/BODY AXES SYSTEM FOR AIRCRAFT MOTION



Ps = P Cos α + R Sin α
Rs = -P Sin α + R Cos α
Angular rates (Stability axes)
Air-path axes forces
Fxw
Fyw
Fzw
α̇ = (Q Cos β - Ps Sin β + Fzw/mVt)/Cos β
β̇ = Fyw/mVt - Rs
V̇t = Fxw/m
Angle of Attack
Angle of Sideslip
True Airspeed
Fxb = Xa Cos α - Za Sin α + Xp
Fyb = Ya + Yp
Fzb = Xa Sin α + Za Cos α + Zp
Body axes forces
Ub = Vt Cos α Cos β
Vb = Vt Sin β
Wb = Vt Sin α Cos β
Velocity components (body-axes)
Wind components
Vwn Vwe Vwd
Ax = Fxb/mg
Ay = Fyb/mg
Az = Fzb/mg
Linear accelerations
Vne = L1 Ub + M1 Vb + N1 Wb - Vwn
Vee = L2 Ub + M2 Vb + N2 Wb - Vwe
Vde = L3 Ub + M3 Vb + N3 Wb - Vwd
Velocity components (Earth-axes)
Ẋ = Vne
Ẏ = Vee
ALṪ = -Vde
Positional coordinates
RANGE = (X² + Y²)^½
RANGE
Euler Angles
Vgr = (Une² + Vee²)^½
γ = θ - α
λ = Tan⁻¹[-Vde/Vgr]
χ = Tan⁻¹[Vee/Vne]
Ground speed
Flight path angle
Climb angle
Angle of track

292

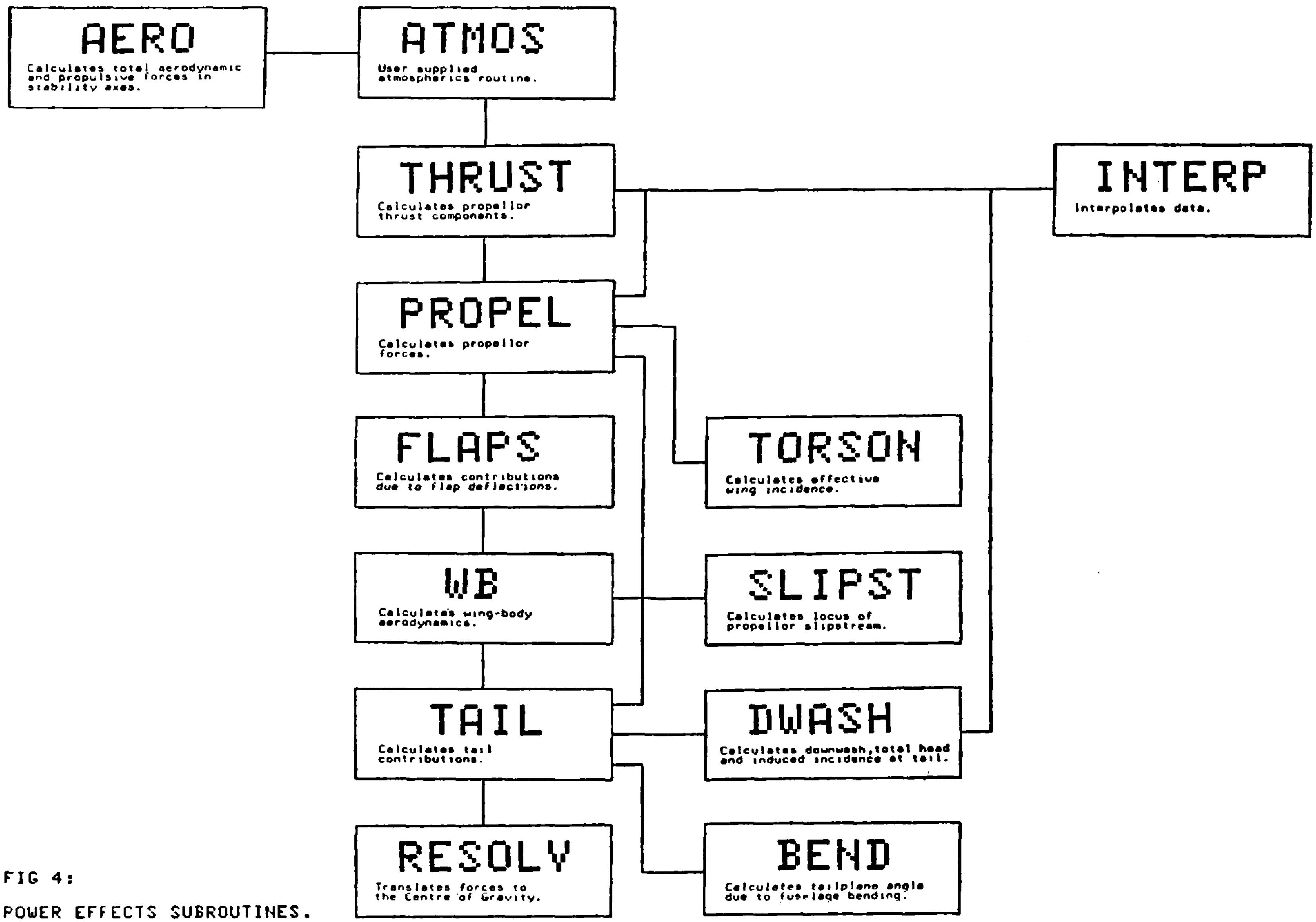


FIG 4:

POWER EFFECTS SUBROUTINES.

APPENDICES
##########

APPENDIX 1. SDOFAP PROGRAM LISTING

SDOFAP.ACSL

```
PROGRAM SDOFAP (FLIGHT SIMULATION MODEL)

     "--------------------------------------------------------------"
     "                                                              "
     " PROGRAM NAME : Six Degrees of Freedom in Air Path axes.      "
     " WRITTEN BY   : P.W.GIBBENS  (EO1, A.R.L., ABS-FW)            "
     " COMPLETED    :    8   MAY      1985                          "
     " DESCRIPTION  :  THIS PROGRAM PRESENTS A SET OF FLIGHT DYNAMIC "
     " EQUATIONS (SIX DEGREES OF FREEDOM), IN AIR-PATH AXES, FOR    "
     " AIRCRAFT SIMULATION USING ADVANCED CONTINUOUS SIMULATION     "
     " LANGUAGE (ACSL) ON THE A.R.L. ELXSI 64ØØ COMPUTER.           "
     "                                                              "
     "--------------------------------------------------------------"


       INTEGER   ITLIM,MAXRES
       LOGICAL   BEGIN
       ARRAY     VECIN(2Ø),XH(3)

       "********************** INITIAL SECTION **********************"

INITIAL

       "*********** PREPARE ALL DATA AND RUNTIME PARAMETERS ***********"

       "*** READ IN CONFIGURATION AND FLIGHT CONDITION DATA ***"

       PROCEDURAL  (PHIDØ,ALTØ,VEØK,IXX,IYY,IZZ,IXZ,MASS=)

       CALL CAFCDI  (PHIDØ,ALTØ,VEØK,IXX,IYY,IZZ,IXZ,MASS)

END    $"OF CAFCDI PROCEDURAL"

       "*** DEFINE ALL PRESET VARIABLES ***"

       "*** RUNTIME PARAMETERS ***"

       CONSTANT             $"#### RUNTIME CONTROL PARAMETERS SHOULD ####"
       CONSTANT             $"####   BE DEFINED AT THIS POINT. SEE   ####"
       CONSTANT             $"####     APPLICATION FOR EXAMPLES.     ####"

       "*** TRIMMING ROUTINE ITERATION PARAMETERS ***"

       CONSTANT             $"#### TRIMMING ROUTINE PARAMETERS SHOULD ###"
                            $"####   BE DEFINED AT THIS POINT. SEE    ###"
                            $"####     APPLICATION FOR EXAMPLES.      ###"
```

```
"*** ATMOSPHERIC STANDARDS AND CONVERSION FACTORS ***"

CONSTANT   RHOØ = 1.2256


"GENERAL CONVERSION FACTORS, CONSTANTS AND VARIABLES"

CONSTANT   DEGTOR= Ø.Ø17453 ,     RADTOD= 57.29578          ...
          ,KTOMPS= Ø.514773 ,     MPSTOK= 1.9426Ø2          ...
          ,MTONM = Ø.ØØØ538 ,     NMTOM = 1858.5            ...
          ,FTTOM = Ø.3Ø48ØØ ,        G  = 9.8Ø7             ...
          ,  PI  = 3.141593


"...WHERE  DEGTOR= DEGREES TO RADIANS               ...
           RADTOD= RADIANS TO DEGREES               ...
           KTOMPS= KNOTS TO METRES PER SECOND       ...
           MPSTOK= METRES PER SECOND TO KNOTS       ...
           MTONM = METRES TO NAUTICAL MILES         ...
           NMTOM = NAUTICAL MILES TO METRES         ...
           FTTOM = FEET TO METRES "


"*** INITIAL CONDITIONS (FLIGHT DATA) ***"

CONSTANT   DPSIDØ =Ø.Ø     ,DPHIDØ =Ø.Ø       ,DALTØ =Ø.Ø  ...
          ,   VWN =Ø.Ø     ,   VWE =Ø.Ø       ,  VWD =Ø.Ø  ...
          , DVEØK =Ø.Ø     ,  DXØ  =Ø.Ø       , DYØ  =Ø.Ø  ...
          , PSIDØ =Ø.Ø     ,   XØ  =Ø.Ø       ,  YØ  =Ø.Ø  ...


"*** ALLOW PARAMETER INCREMENTING ***"

           "#### INCREMENTS TO FLIGHT CONDITION PARAMETERS ####"
           "####  SHOULD BE DEFINED AT THIS POINT. THOSE    ####"
           "#### SHOWN PERTAIN TO THE EXAMPLE APPLICATION.  ####"

CONSTANT   DEL1 =Ø.Ø     ,  DRPM =Ø.Ø       , DXCGP=Ø.Ø ...
          ,DPL  =Ø.Ø     ,  DEL5 =Ø.Ø       , DEL6 =Ø.Ø


"*** SET CONTROL INPUT PARAMETERS ***"

           "#### REQUIRED CONTROL INPUT PARAMETERS SHOULD ####"
           "####  BE DEFINED AT THIS POINT. THOSE SHOWN   ####"
           "####   PERTAIN TO THE EXAMPLE APPLICATION.    ####"

CONSTANT ESTART=Ø.Ø1,TSTART=Ø.Ø1 $"** PULSE START TIME **"
CONSTANT EPULSE=5.Ø ,TPULSE=12Ø. $"** PULSE DURATION   **"
CONSTANT EREPET=2ØØ.,TREPET=2ØØ. $"** REPETITION TIME  **"
CONSTANT ETAMAX=5.Ø ,THRMAX=1.Ø  $"** PULSE AMPLITUDE  **"

   PULMAX =ETAMAX*DEGTOR         $"** CONVERT TO RADIANS **"


"************** INCREMENT DATA AND PARAMETERS ****************"


"*** ADD RUNTIME INCREMENTS TO FLIGHT DATA ***"

   PSIDØ =PSIDØ +DPSIDØ
   PHIDØ =PHIDØ +DPHIDØ
   ALTØ  =(ALTØ +DALTØ)*FTTOM
```

```
      VEØK   =VEØK   +DVEØK
      XØ     =XØ     +DXØ
      YØ     =YØ     +DYØ
      GAMMAR=GAMMAR+DGAMMR

   "*** DEFINE INITIAL Z COORDINATE. ***"

      ZØ =-ALTØ

   "*** ADD RUNTIME INCREMENTS TO PARAMETERS ***"

      VECIN(1)=ALTØ    $"#### INCREMENTS OF FLIGHT CONDITION DATA ####"
      VECIN(2)=DRPM    $"####   ARE ADDED AT SUBROUTINE LEVEL.     ####"
      VECIN(3)=DXCGP
      VECIN(4)=DPL     $"#### INITIAL ALTITUDE IS ALSO REQUIRED ####"
      VECIN(5)=DEL5    $"####        AT SUBROUTINE LEVEL.        ####"
      VECIN(6)=DEL6

   CALL PARINC(VECIN,CGPOS,PLS)

   "*** CONVERT FROM EAS TO TAS ***"

   PROCEDURAL  (RHO=ALTØ)

   CALL ATMOS  (ALTØ,RHO)

END   $" OF ATMOS PROCEDURAL"

      VTØK=VEØK*SQRT (RHOØ/RHO)

      VEØ  =  VEØK*KTOMPS
      VTØ  =  VTØK*KTOMPS

   "************* SPECIFY SYSTEM EXCITATION PARAMETERS ************"

   "*** SPECIFY TERMINATION CONDITION ***"

   CONSTANT   TSTOP =Ø.Ø

   "*** SPECIFY INDEPENDENT VARIABLE AS A PRECAUTION ***"

   VARIABLE     TIME = Ø.Ø
   CINTERVAL    CINT = Ø.Ø
   NSTEPS       NSTP = Ø

   "******* APPROXIMATE TRIM VALUES FOR INITIAL CONDITIONS *******"

      PSIRØ = PSIDØ*DEGTOR
      PHIRØ = PHIDØ*DEGTOR

   PROCEDURAL  (PØ,QØ,RØ,BETARØ,ALPHRØ,THETRØ,ETARØ      ...
                ,GAMMAR=VTØ,PHIRØ,G)

   CALL TRAP  (VTØ,PHIRØ,G,GAMMAR,PØ,QØ,RØ,BETARØ        ...
               ,ALPHRØ,THETRØ,ETARØ)

END   $" OF TRAP PROCEDURAL "
```

```
      "*** CALCULATE INERTIAL CONSTANTS ***"

         CØ  = ((IXX*IZZ)-(IXZ*IXZ))
         C1  = IZZ/CØ
         C2  = IXZ/CØ
         C3  = C2*(IXX-IYY+IZZ)
         C4  = ((IYY-IZZ)*C1)-(IXZ*C2)
         C5  = 1.Ø/IYY
         C6  = C5*IXZ
         C7  = C5*(IZZ-IXX)
         C8  = IXX/CØ
         C9  = ((IXX-IYY)*C8)+(IXZ*C2)

      "***************** INITIALIZE QUATERNIONS ********************"

      THETDØ= THETRØ*RADTOD

1Ø .. CONTINUE

      "**NOTE:-THE EULER ANGLE(S) ARE HALVED WHEN CALCULATING...
              THE QUATERNIONS"

      CALL QUATNS(PSIRØ,THETRØ,PHIRØ,TAUØØ,TAU1Ø,TAU2Ø,TAU3Ø)

END $"OF INITIAL"


      "********************* DYNAMIC SECTION ***********************"

DYNAMIC

      "*** THE TRANSITION FROM INITIAL TO DYNAMIC TRANSFERS    ***"
      "*** ALL INITIAL CONDITIONS TO THE STATE VARIABLES AND   ***"
      "*** EVALUATES THE CODE IN THE DERIVATIVE SECTION ONCE.  ***"

      IF(BEGIN) GO TO 2Ø

      "********* TRIM AIRCRAFT WITH USER SUPPLIED SUBROUTINE ********"

              "#### EXAMPLE OF TRIMMING IN RECTILINEAR ####"
              "####   FLIGHT USING SUBROUTINE POWIT.   ####"

         XH(1) =THETRØ      $"#### OTHER TRIM CONDITIONS MAY BE      ####"
         XH(2) =ALPHRØ      $"#### OBTAINED BY DEFINING APPROPRIATE ####"
         XH(3) =ETARØ       $"#### TRIM STATES AND ASSOCIATED TRIM   ####"
                            $"#### CONTROLS. EG. LEVEL BANKED TURN.  ####"

      PROCEDURAL(XH=XH,MAXRES,ITLIM,ERRMAX)

      CALL POWIT(XH,MAXRES,ERRMAX,ITLIM)

         THETRØ =XH(1)
         ALPHRØ =XH(2)
         ETARØ  =XH(3)

END    $ "OF POWIT PROCEDURAL"
```

```
      BEGIN=.TRUE.
      GO TO 1Ø

2Ø .. CONTINUE


      "********************* DERIVATIVE SECTION *********************"

DERIVATIVE


      "****************** CALCULATE CONTROL INPUTS ******************"


         "#### EXAMPLE OF CONTROL INPUTS USING THE ACSL PULSE ####"
         "####   GENERATOR, OTHER INPUT FORMS ARE AVAILABLE.  ####"

      PROCEDURAL(ETAR,PLS=                                   ...
      ETARØ,DELPL,ESTART,EREPET,EPULSE,TSTART,TREPET,TPULSE)

         DELETA=PULMAX*PULSE(ESTART,EREPET,EPULSE)
         DELPL =THRMAX*PULSE(TSTART,TREPET,TPULSE)
         ETAR=ETARØ+DELETA

      CALL CONTROLS(ETAR,DELPL,PLS)

         ETAD=ETAR*RADTOD

END   $ "OF CONTROLS PROCEDURAL"


      "********* CALCULATE QUATERNIONS; NORMALIZE, AND THEN *********"
      "*********        DETERMINE THE DIRECTION COSINES      *********"

      PROCEDURAL(L1,L2,L3,M1,M2,M3,N1,N2,N3=TAUØ,TAU1,TAU2,TAU3)


      "*** NORMALIZE QUATERNIONS ***"

         TAUN  = SQRT((TAUØ*TAUØ)+(TAU1*TAU1)+(TAU2*TAU2)+(TAU3*TAU3))
         TAUØN = TAUØ/TAUN
         TAU1N = TAU1/TAUN
         TAU2N = TAU2/TAUN
         TAU3N = TAU3/TAUN

      "*** CALCULATE THE DIRECTIONAL COSINES (L1 -> N3) ***"

         L1    = (((TAUØN*TAUØN)+(TAU1N*TAU1N))*2.Ø)-1.Ø
         L2    =  ((TAU1N*TAU2N)+(TAUØN*TAU3N))*2.Ø
         L3    =  ((TAU1N*TAU3N)-(TAUØN*TAU2N))*2.Ø

         M1    =  ((TAU1N*TAU2N)-(TAUØN*TAU3N))*2.Ø
         M2    = (((TAUØN*TAUØN)+(TAU2N*TAU2N))*2.Ø)-1.Ø
         M3    =  ((TAU2N*TAU3N)+(TAUØN*TAU1N))*2.Ø

         N1    =  ((TAU1N*TAU3N)+(TAUØN*TAU2N))*2.Ø
         N2    =  ((TAU2N*TAU3N)-(TAUØN*TAU1N))*2.Ø
         N3    = (((TAUØN*TAUØN)+(TAU3N*TAU3N))*2.Ø)-1.Ø

END $"OF QUATERNION PROCEDURAL"
```

```
        "*** CALCULATE ATMOSPHERIC CONDITIONS FOR CURRENT ALTITUDE ***"

         CALL ATMOS (-Z,RHO)   $"#### ATMOSPHERICS SUBROUTINE MUST ####"
                                 "####   BE SUPPLIED BY THE USER.   ####"

        "*** CALCULATE LONGITUDINAL AND LATERAL AERODYNAMIC FORCES ***"
        "********* IN STABILITY AXES AND MOMENTS IN BODY AXES *********"

PROCEDURAL (XA,YA,ZA,LA,MA,NA=VT,ALPHAR,BETAR,P,Q,R,PHIR,THETAR,DALPHR)

        "#### THE AERO SUBROUTINE SUPPLIES THE AERODYNAMIC FORCES ####"
        "####  IN STABILITY AXES. THIS MUST BE SUPPLIED BY USER.  ####"

        CALL AERO (XA,YA,ZA,LA,MA,NA,VT,ALPHAR,BETAR,P,Q,R,   ...
                   PHIR,THETAR,DALPHR)

END $"OF AERO PROCEDURAL"

PROCEDURAL (XP,YP,ZP,LP,MP,NP=VT,ALPHAR,BETAR,P,Q,R,PHIR,THETAR)

        "#### THE PROP SUBROUTINE SUPPLIES THE PROPULSIVE FORCES  ####"
        "####    IN BODY AXES. THIS MUST BE SUPPLIED BY USER.     ####"

        CALL PROP (XP,YP,ZP,LP,MP,NP,VT,ALPHAR,BETAR,P,Q,R,   ...
                   PHIR,THETAR)

END $"OF PROP PROCEDURAL"

           SA = SIN(ALPHAR)
           CA = COS(ALPHAR)
           SB = SIN(BETAR)
           CB = COS(BETAR)

        "*** CALCULATE GRAVITY FORCE COMPONENTS -> STABILITY AXES ***"

           FGX = L3*MASS*G*CA + N3*MASS*G*SA
           FGY = M3*MASS*G
           FGZ =-L3*MASS*G*SA + N3*MASS*G*CA

        "*** CALCULATE THE TOTAL FORCES (FXS,FYS,FZS) ***"
        "***             -> STABILITY AXES            ***"

           FXS = XA + XP*CA + ZP*SA + FGX
           FYS = YA + YP            + FGY
           FZS = ZA - XP*SA + ZP*CA + FGZ

        "*** CALCULATE THE TOTAL MOMENTS (L,M,N) -> BODY AXES ***"

           L   = (LA*CA) - (NA*SA) + LP
           M   = MA                + MP
           N   = (LA*SA) + (NA*CA) + NP

        "*** CALCULATE THE TOTAL FORCES (FXW,FYW,FZW) ***"
        "***             -> AIR-PATH AXES             ***"

           FXW = FXS*CB + FYS*SB
           FYW =-FXS*SB + FYS*CB
           FZW = FZS
```

```
        "*** CALCULATE FORCES IN BODY AXES ***"

           FXB = XA*CA - ZA*SA + XP
           FYB = YA + YP
           FZB = XA*SA + ZA*CA + ZP


        "*** CALCULATE LINEAR ACCELERATIONS ***"

           AX  =  FXB/(MASS*G)
           AY  =  FYB/(MASS*G)
           AZ  =  FZB/(MASS*G)


        "*** CALCULATE NORMAL ACCELERATION ***"

           AN  = -AZ


        "*** RESOLVE BODY AXIS ANGULAR RATES INTO STABILITY AXES ***"

           PS  =  P*CA + R*SA
           RS  = -P*SA + R*CA


        "****** CALCULATE LINEAR DERIVATIVES -> AIR PATH AXES ******"

        "*** DECLARE DALPHR AS AN IMPLICIT VARIABLE ***"

           DALPHR = IMPL (Q,Ø.ØØ1,3Ø,EF,                          ...
                         (Q*CB-PS*SB+FZW/MASS/VT)/CB,Ø.ØØØ1)

PROCEDURAL (=EF)

        CALL IMP(EF)

END $"OF DALPHR PROCEDURAL"


           DBETAR = FYW/MASS/VT-RS
           DVT    = FXW/MASS


        "*** CALCULATE ANGULAR ACCELERATIONS -> BODY AXES ***"

           PDOT = (L*C1)+(N*C2)+(((P*C3)+(R*C4))*Q)
           QDOT = (M*C5)+(((R*R)-(P*P))*C6)+(R*P*C7)
           RDOT = (N*C8)+(L*C2)+(((P*C9)-(R*C3))*Q)


        "******** DETERMINE QUATERNION RATES AND EULER ANGLES. ********"


        "*** CALCULATE THE QUATERNION RATES ***"

           TAUØDT=-(((TAU1N*P)+(TAU2N*Q)+(TAU3N*R))*Ø.5)
           TAU1DT=  ((TAUØN*P)-(TAU3N*Q)+(TAU2N*R))*Ø.5
           TAU2DT=  ((TAU3N*P)+(TAUØN*Q)-(TAU1N*R))*Ø.5
           TAU3DT=-(((TAU2N*P)-(TAU1N*Q)-(TAUØN*R))*Ø.5)


        "*** CALCULATE THE NEW EULER ANGLES ***"

PROCEDURAL(PSID,THETAD,PHID=L1,L2,L3,M3,N3)
```

```
      "*** CALCULATE HEADING (OR YAW) ANGLE (PSI) ***"
      "***        IN RANGE Ø(NORTH) TO 36Ø DEG.     ***"

         PSIR = ATAN(L2/L1)
         IF(PSIR.LT.Ø.Ø) PSIR = PSIR+PI*2
         PSID = PSIR*RADTOD


      "*** CALCULATE ANGLE OF PITCH (THETA) ***"
      "***        IN RANGE +/-9Ø DEGREES     ***"

         THETAR= ATAN(-L3/SQRT((L1*L1)+(L2*L2)))
         THETAD= THETAR*RADTOD


      "***   CALCULATE BANK (OR ROLL) ANGLE (PHI) IN RANGE   ***"
      "*** +/-18Ø DEGREES,WHERE Ø DEG. INDICATES WINGS LEVEL ***"

         PHIR = ATAN(M3/N3)
         PHID = PHIR*RADTOD

END $"OF EULER PROCEDURAL"

       "***********************  TRAJECTORY  ***********************"


       "***      RESOLVE FLIGHT PATH AXES VELOCITIES INTO BODY      ***"
       "*** COMPONENTS FOR CALCULATION OF DISTANCES IN EARTH AXES ***"

          UB = VT*CA*CB
          VB = VT*SB
          WB = VT*SA*CB


       "***      CALCULATE EARTH AXES VELOCITIES FROM BODY AXES      ***"
       "*** VELOCITIES USING DIRECTION COSINES AND WIND VELOCITIES ***"

          VNE = L1*UB+M1*VB+N1*WB-VWN
          VEE = L2*UB+M2*VB+N2*WB-VWE
          VDE = L3*UB+M3*VB+N3*WB-VWD


PROCEDURAL(VGRKT,GAMMAD,CHID=VNE,VEE,VDE)


       "*** CALCULATE THE VELOCITY OVER THE GROUND, ***"
       "***          VGR (IE. GROUND SPEED)         ***"

          VGR   = SQRT((VNE*VNE)+(VEE*VEE))
          VGRKT = VGR*MPSTOK


      "*** CALCULATE FLIGHT-PATH ANGLE, GAMMA IN RANGE +/-9Ø DEG ***"

         GAMMAR = THETAR-ALPHAR
         GAMMAD = GAMMAR*RADTOD


      "*** CALCULATE ANGLE OF CLIMB, LAMBDA IN RANGE +/-9Ø DEG ***"

         LAMDAR = ATAN(-VDE/VGR)
         LAMDAD = LAMDAR*RADTOD
```

```
      "*** CALCULATE ANGLE OF TRACK, CHI, ***"
      "*** IN RANGE Ø(NORTH) TO 36Ø DEG.  ***"

         CHIR   = ATAN(VEE/VNE)
         CHID   = CHIR*RADTOD

END $"OF TRAJECTORY PROCEDURAL"

      "*** CALCULATE THE RANGE (NOTE: IN METRES) ***"

         RANGE  = SQRT((X*X)+(Y*Y))

      "*********** INTEGRATION OF SYSTEM STATE EQUATIONS ***********"

         ALPHAR = INTVC(DALPHR,ALPHRØ)
         BETAR  = INTVC(DBETAR,BETARØ)
         VT     = INTVC(DVT,VTØ)

         P      = INTVC(PDOT,PØ)             $"ROLL RATE"
         Q      = INTVC(QDOT,QØ)             $"PITCH RATE"
         R      = INTVC(RDOT,RØ)             $"YAW RATE"

         TAUØ   = INTVC(TAUØDT,TAUØØ)        $"QUATERNION TERMS"
         TAU1   = INTVC(TAU1DT,TAU1Ø)
         TAU2   = INTVC(TAU2DT,TAU2Ø)
         TAU3   = INTVC(TAU3DT,TAU3Ø)

         Z      = INTEG(VDE,ZØ)              $"Z-POSITIONAL CO-ORDINATE"
         X      = INTEG(VEE,XØ)              $"X-POSITIONAL CO-ORDINATE"
         Y      = INTEG(VNE,YØ)              $"Y-POSITIONAL CO-ORDINATE"

END $"OF DERIVATIVE"

         VE     = VT*SQRT(RHO/RHOØ)          $"EQUIVALENT AIRSPEED (M/S)"
         VEK    = VE*MPSTOK                  $"EQUIVALENT AIRSPEED (KTS)"
         ALPHAD = ALPHAR*RADTOD              $"ANGLE OF ATTACK     (DEG)"
         ALT    =-Z/FTTOM                    $"ALTITUDE            (FT.)"

      "*************** EXPRESS TERMINATION CONDITION ***************"

      TERMT(TIME.GE.TSTOP)

END $"OF DYNAMIC"

      "*** NOTE:   THIS LISTING SHOULD BE USED IN CONJUNCTION WITH ***"
      "*** THE ADVANCED CONTINUOUS SIMULATION LANGUAGE (ACSL) USER ***"
      "*** GUIDE/REFERENCE MANUAL.                                 ***"

END $"OF PROGRAM"
```

```
      SUBROUTINE QUATNS(PSIR,THETAR,PHIR,TAUØ,TAU1,TAU2,TAU3)

C     "*** CALCULATES NORMALIZED QUATERNIONS ***"

         SPSI   = SIN(PSIR*Ø.5)
         CPSI   = COS(PSIR*Ø.5)
         STHETA = SIN(THETAR*Ø.5)
         CTHETA = COS(THETAR*Ø.5)
         SPHI   = SIN(PHIR*Ø.5)
         CPHI   = COS(PHIR*Ø.5)


C     "*** CALCULATE THE QUATERNIONS FROM THE EULER ANGLE TERMS ***"

         TAUØ = (CPHI*CTHETA*CPSI)+(SPHI*STHETA*SPSI)
         TAU1 = (SPHI*CTHETA*CPSI)-(CPHI*STHETA*SPSI)
         TAU2 = (CPHI*STHETA*CPSI)+(SPHI*CTHETA*SPSI)
         TAU3 = (CPHI*CTHETA*SPSI)-(SPHI*STHETA*CPSI)


C     "*** NORMALIZE INITIAL QUATERNIONS ***"

         TAUN = SQRT((TAUØ*TAUØ)+(TAU1*TAU1)+(TAU2*TAU2)
     1          +(TAU3*TAU3))
         TAUØ = TAUØ/TAUN
         TAU1 = TAU1/TAUN
         TAU2 = TAU2/TAUN
         TAU3 = TAU3/TAUN


      RETURN

      END



      SUBROUTINE EVAL (XXXX,XXXRES,MAXR)

C     "*** THIS SUBROUTINE IS CALLED FROM THE TRIMMING SUBROUTINE ***"
C     "***          TO GAIN ACCESS TO THE DERIVATIVE SECTION.       ***"

      DIMENSION XXXX(3),XXXRES(3)
$
C     "*** THE "$" CONTROL CHARACTER MAKES MAIN PROGRAM ***"
C     "***    VARIABLES AVAILABLE TO THIS SUBROUTINE.   ***"

      INTEGER I
         I=1

         THETAR = (XXXX(1))
         ALPHAR = (XXXX(2))
         ETARØ  = (XXXX(3))

      CALL QUATNS(PSIRØ,THETAR,PHIRØ,TAUØ,TAU1,TAU2,TAU3)

      CALL ZZDERV(I)

         XXXRES(1) = (DVT)
         XXXRES(2) = (DALPHR)
         XXXRES(3) = (QDOT)

      RETURN
```

```
      END
      SUBROUTINE IMP(EF)
C     "***** NOTIFIES FAILURE OF IMPLICIT ITERATION ROUTINE *****"

      IF (EF .EQ. Ø.Ø) GOTO 89
      WRITE (6,99)
99    FORMAT(//27HIMPLICIT PROCEDURAL FAILED   //)
89    RETURN
      END
```

FTSUBS.F

```
      SUBROUTINE CAFCDI(PHIN,HNM,VN,IXX,IYY,IZZ,IXZ,MASS)

      IMPLICIT REAL(A-Z)
      INCLUDE 'FTPAR.F'
      INTEGER I,J

C     ****** ROUTINE FOR THE INPUT OF CONFIGURATION AND ******
C     *************** FLIGHT CONDITION DATA. ****************

      OPEN (UNIT=1,FILE='FTSUD2.IN',STATUS='OLD')
      OPEN (UNIT=7,FILE='FTGRAF.VPP',STATUS='OLD')

C     *** READ IN CONFIGURATION AND FLIGHT CONDITION DATA ***

C     #### THE VARIABLES SHOWN PERTAIN TO THE EXAMPLE APPLICATION  ####
C     #### DATA READ IN IS AVAILABLE AT SUBROUTINE LEVEL ONLY.     ####

      READ(1,*)TTOT,TSAMP,NH,HN,DELH,NV,VN,DELV,NPHI,PHIN
      READ(1,*)DELPHI,NPL,PLØ,DELPL,NRPM,RPM,DELRPM,WEGHT,NXCG,XCGP
      READ(1,*)DELXCG,ZCG,(PEFF(I),I=1,7),XT
      READ(1,*)ZT,XTH,ZTH,THSET,MAXEP,NBLAD,PDIA,WSET,TPSET,ETAG
      READ(1,*)ACMASS,ACIXX,ACIYY,ACIZZ,ACIXZ,SW,CW,BW,ST,CT
      READ(1,*)CETA,XP,ZP,CWP,BFW,BTAIL,CLØ,CLAL,CDØ,CDAL
      READ(1,*)CMØ,CMAL,EPSØ,EPSAL,QTOQ,AØ,A1,A2,A3,BØ
      READ(1,*)B1,B2,B3,CDØT,CDLT,CMTØ,CMQW,CLP,CLXI,KWING
      READ(1,*)KFUSE,NPSFI,NPSFR,MPSFI,MPSFR,CDB,GTR,BLØ,WTR,TAPERF
      READ(1,*)XWC,ZWC,SØS,XQARTC

      READ(7,*) (FPROP(I,1),FPROP(I,2),I=1,2Ø)
      READ(7,*) ((TOP(I,J),J=1,33),I=1,2Ø)
      READ(7,*) ((ETP(I,J),J=1,6Ø),I=1,21)
111   READ(7,*) (CYPROP(I,1),CYPROP(I,2),CYPROP(I,3),CYPROP(I,4),I=1,2Ø)
      READ(7,*) ((DELEPS(I,J),J=1,1Ø),I=1,8)

C     #### DATA WHICH IS REQUIRED IN THE MAIN PROGRAM MUST ####
C     ####    RENAMED BEFORE BEING PASSED AS ARGUMENTS.    ####

      MASS=ACMASS
      IXX=ACIXX
      IYY=ACIYY
      IZZ=ACIZZ
      IXZ=ACIXZ

      HNM=HN*.3Ø48

      CLOSE(UNIT=1)
      CLOSE(UNIT=7)

      RETURN

      END
```

```
      SUBROUTINE PARINC (VECIN,CGPOS,PLS)

C     "*** SUBROUTINE TO ALLOW RUNTIME PARAMETER MODIFICATIONS ***"

      IMPLICIT REAL(A-Z)
      DIMENSION VECIN(6)
      INCLUDE 'FTPAR.F'

C     #### THE VARIABLES SHOWN PERTAIN TO THE EXAMPLE APPLICATION. ####
C     #### THEY DEMONSTRATE THE METHOD OF PARAMETER INCREMENTING.  ####

      ALTØ=      VECIN(1)
      RPM =RPM +VECIN(2)
      XCGP=XCGP+VECIN(3)
      PLØ =PLØ +VECIN(4)
      PAR5=PAR5+VECIN(5)
      PAR6=PAR6+VECIN(6)

      XCG=XCGP*CW/1ØØ.+XQARTC-CW/4.

C     #### DATA WHICH IS REQUIRED IN THE MAIN PROGRAM MUST ####
C     ####    RENAMED BEFORE BEING PASSED AS ARGUMENTS.    ####

      CGPOS=XCGP
      PLS=PLØ

      RETURN

      END


      SUBROUTINE TRAP(VTØ,PHIRØ,G,GAMMAR,PØ,QØ,RØ,BETARØ
     1                ,ALPHRØ,THETRØ,ETARØ)

C     ******** GIVES INITIAL APPROXIMATION FOR TRIM CONDITION *********

C     #### THE APPROXIMATION SHOWN PERTAINS TO THE   ####
C     #### EXAMPLE APPLICATION. THE APPROXIMATION TO ####
C     #### BE USED IS AT THE USERS DISCRETION.       ####

      IMPLICIT REAL(A-Z)
      INCLUDE 'FTPAR.F'

      PI = 3.141593
      ETAØØ= 2.2
      KK =41.Ø*(XCGP/1ØØ.-Ø.4)
      QD =(.5*1.2256*VTØ*VTØ)

      PØ =Ø.
      QØ =G/VTØ*TAN(PHIRØ)*SIN(PHIRØ)
      RØ =G/VTØ*SIN(PHIRØ)

      AN  = 1/COS(PHIRØ)

      BETARØ = Ø.


      ALPHAW = ACMASS*G/(CLAL*QD*SW)
      ALPHRØ = ALPHAW - WSET
      CLWB   = CLØ + (CLAL*ALPHAW)
      CDWB   = CDØ + (CDAL*CLWB**2)

      DRAG   = CDWB * QD * SW

      GAMMAR = ASIN(-DRAG/(ACMASS*G))
      THETRØ = GAMMAR + ALPHRØ
```

```
      ETAØ    = (ETAØØ+KK*CLWB)*PI/18Ø.
      ETARØ   = ETAØ



      RETURN

      END


      SUBROUTINE ATMOS (H,RO)

      IMPLICIT REAL(A-Z)

      COMMON/ARGS2/HITE,DENS

C     **** THIS ROUTINE GIVES THE DENSITY AT A ****
C     **** GIVEN ALTITUDE FOR ISA CONDITIONS.   ****

C     #### MORE COMPLICATED ATMOSPHERE ROUTINES MAY BE USED.       ####
C     #### THE SUBROUTINE TO BE USED IS AT THE USERS DISCRETION. ####


      HITE=H
      DENS=1.2256*(1-Ø.Ø2256*HITE/1ØØØ.Ø)**4.2561
      RO  =DENS


      RETURN

      END


      SUBROUTINE AERO (XA,YA,ZA,LA,MA,NA,VT,ALPHAR,BETAR
     1                  ,PP,QQ,RR,PHIR,THETAR)

C     ******** CALCULATES TOTAL AERODYNAMIC FORCES AND MOMENTS ********

      IMPLICIT REAL(A-Z)
      INCLUDE 'FTPAR.F'
      DIMENSION X(2Ø)


C     #### THE VECTOR X IS USED TO PASS STATE VARIABLES TO   ####
C     #### ANY SUBROUTINES CALLED FROM AERO. SEE THE EXAMPLE ####
C     ####          APPLICATION FOR DETAILS OF ITS USE.       ####

      X(V) = VT
      X(ALPHA) = ALPHAR
      X(Q) = QQ
      X(P) = PP
      X(H) = HITE
      X(PHI) = PHIR
      X(THETA) = THETAR

      CALL ATMOS(HITE,RHO)

      QD=.5*RHO*VT**2

C     ******** DETERMINE INDIVIDUAL AERODYNAMIC CONTRIBUTIONS. ********


C     #### ANY SUBROUTINE CALLS OR OTHER CALCULATIONS ####
```

```
C     #### TO DETERMINE INDIVIDUAL AERODYNAMIC FORCE  ####
C     #### CONTRIBUTIONS SHOULD BE ENTERED HERE.      ####
C     #### SEE APPLICATION FOR EXAMPLES.              ####

C     *******  CALCULATE TOTAL AERODYNAMIC FORCES AND MOMENTS. ********

C     #### TOTAL FORCES AND MOMENTS SHOULD BE DETERMINED ####
C     #### HERE BY ADDITION OF INDIVIDUAL CONTRIBUTIONS. ####
C     #### SEE APPLICATION FOR EXAMPLES.                 ####

      XA =
      YA =
      ZA =

      LA =
      MA =
      NA =



      RETURN
      END


      SUBROUTINE PROP(XP,YP,ZP,LP,MP,NP,VT,ALPHAR,BETAR,
     1                     PP,QQ,RR,PHIR,THETAR)

C     ******** CALCULATES TOTAL PROPULSIVE FORCES AND MOMENTS ********

      IMPLICIT REAL(A-Z)
      INCLUDE 'FTPAR.F'
      DIMENSION X(2Ø)


C     #### THE VECTOR X IS USED TO PASS STATE VARIABLES TO   ####
C     #### ANY SUBROUTINES CALLED FROM PROP. SEE THE EXAMPLE ####
C     ####         APPLICATION FOR DETAILS OF ITS USE.       ####

      X(V) = VT
      X(ALPHA) = ALPHAR
      X(Q) = QQ
      X(P) = PP
      X(H) = HITE
      X(PHI) = PHIR
      X(THETA) = THETAR

      CALL ATMOS(HITE,RHO)

      QD=.5*RHO*VT**2

C     ******** DETERMINE INDIVIDUAL PROPULSIVE CONTRIBUTIONS. ********


C     #### ANY SUBROUTINE CALLS OR OTHER CALCULATIONS ####
C     #### TO DETERMINE PROPULSIVE FORCES SHOULD BE   ####
C     #### ENTERED HERE.                              ####
C     #### SEE APPLICATION FOR EXAMPLES.              ####


C     *******  CALCULATE TOTAL PROPULSIVE FORCES AND MOMENTS. ********

C     #### TOTAL FORCES AND MOMENTS SHOULD BE DETERMINED ####
C     #### HERE BY ADDITION OF INDIVIDUAL CONTRIBUTIONS. ####
C     #### SEE APPLICATION FOR EXAMPLES.                 ####

      XP =
```

```
      YP =
      ZP =

      LP =
      MP =
      NP =


      RETURN
      END


       SUBROUTINE CONTROLS(ETAR,DELPL,PLS)

       INCLUDE 'FTPAR.F'

C      *** THIS SUBROUTINE IS USED TO INCREMENT OR OTHERWISE   ***
C      *** MODIFY CONTROL INPUTS AS REQUIRED.                  ***

C      #### THE CODING SHOWN RELATES TO THE EXAMPLE APPLICATION. ####

       ETA = ETAR
       PL  = PLØ+DELPL
       PLS = PL

       RETURN

       END
```

APPENDIX 2. POWER EFFECTS PROGRAM LISTING

SDOFAP.ACSL

```
PROGRAM SDOFAP (FLIGHT SIMULATION MODEL)

      "----------------------------------------------------------------"
      "                                                                "
      " PROGRAM NAME : Six Degrees of Freedom in Air Path axes.        "
      " WRITTEN BY   : P.W. GIBBENS  (EO1, ABS-FW, ARL)                "
      " COMPLETED    :    8   MAY     1985                             "
      "                                                                "
      " MODIFIED BY R.H. PERRIN (EO1, ABS-RW, ARL) TO ALLOW FOR        "
      " A VERY SMALL OR ZERO CLIMB ANGLE - OCTOBER 1985                "
      "                                                                "
      " DESCRIPTION  :  THIS PROGRAM PRESENTS A SET OF FLIGHT DYNAMIC  "
      " EQUATIONS (SIX DEGREES OF FREEDOM), IN AIR-PATH AXES, FOR      "
      " AIRCRAFT SIMULATION USING ADVANCED CONTINUOUS SIMULATION       "
      " LANGUAGE (ACSL) ON THE ARL ELXSI 64ØØ COMPUTER.                "
      "                                                                "
      "----------------------------------------------------------------"


       INTEGER   ITLIM,MAXRES
       LOGICAL   BEGIN
       ARRAY     VECIN(2Ø),XH(3)

       "********************** INITIAL SECTION **********************"

INITIAL

       "********** PREPARE ALL DATA AND RUNTIME PARAMETERS **********"

       "*** READ IN CONFIGURATION AND FLIGHT CONDITION DATA ***"

       PROCEDURAL (PHIDØ,ALTØ,VEØK,IXX,IYY,IZZ,IXZ,MASS=)

       CALL CAFCDI (PHIDØ,ALTØ,VEØK,IXX,IYY,IZZ,IXZ,MASS)

END    $"OF CAFCDI PROCEDURAL"

       "*** DEFINE ALL PRESET VARIABLES ***"

       "*** DEFINE RUNTIME PARAMETERS ***"

       CONSTANT   TSTP=5.   ,PLS=Ø.   ,CGPOS=25.1
       CONSTANT   HI1=Ø.    ,HI2=Ø.   ,HI3=Ø.
       CONSTANT   LO1=Ø.    ,LO2=Ø.   ,LO3=Ø.

       "*** POWIT TRIMMING ROUTINE ITERATION PARAMETERS ***"

       CONSTANT   MAXRES =3             ,ITLIM =1ØØ        ...
                 ,ERRMAX =Ø.ØØØØØ1
```

```
"*** ATMOSPHERIC STANDARDS AND CONVERSION FACTORS ***"

CONSTANT   RHOØ = 1.2256

"*** GENERAL CONVERSION FACTORS, CONSTANTS AND VARIABLES ***"

CONSTANT   DEGTOR= Ø.Ø17453 ,    RADTOD= 57.29578          ...
          ,KTOMPS= Ø.514773 ,    MPSTOK= 1.9426Ø2          ...
          ,MTONM = Ø.ØØØ538 ,    NMTOM = 1858.5            ...
          ,FTTOM = Ø.3Ø48ØØ ,      G   = 9.8Ø7             ...
          ,  PI  = 3.141593

"...WHERE  DEGTOR= DEGREES TO RADIANS               ...
           RADTOD= RADIANS TO DEGREES               ...
           KTOMPS= KNOTS TO METRES PER SECOND       ...
           MPSTOK= METRES PER SECOND TO KNOTS       ...
           MTONM = METRES TO NAUTICAL MILES         ...
           NMTOM = NAUTICAL MILES TO METRES         ...
           FTTOM = FEET TO METRES "

"*** INITIAL CONDITIONS (FLIGHT DATA) ***"

CONSTANT   DPSIDØ =Ø.Ø     ,DPHIDØ =Ø.Ø       ,DALTØ =Ø.Ø  ...
          ,   VWN =Ø.Ø     ,   VWE =Ø.Ø       ,  VWD =Ø.Ø  ...
          , DVEØK =Ø.Ø     , DXØ   =Ø.Ø       , DYØ  =Ø.Ø  ...
          , PSIDØ =Ø.Ø     ,  XØ   =Ø.Ø       ,  YØ  =Ø.Ø  ...

"*** ALLOW PARAMETER INCREMENTING ***"

CONSTANT   DEL1 =Ø.Ø     , DRPM =Ø.Ø     , DXCGP=Ø.Ø ...
          ,DPL  =Ø.Ø     , DEL5 =Ø.Ø     , DEL6 =Ø.Ø

"*** SET CONTROL INPUT PARAMETERS ***"

CONSTANT ESTART=Ø.Ø1,TSTART=Ø.Ø1 $"** PULSE START TIME **"
CONSTANT EPULSE=5.Ø ,TPULSE=12Ø. $"** PULSE DURATION   **"
CONSTANT EREPET=2ØØ.,TREPET=2ØØ. $"** REPETITION TIME  **"
CONSTANT ETAMAX=5.Ø ,THRMAX=1.Ø  $"** PULSE AMPLITUDE  **"

   PULMAX =ETAMAX*DEGTOR         $"*** CONVERT TO RADIANS ***"

"*** ADD RUNTIME INCREMENTS TO FLIGHT DATA ***"

   PSIDØ =PSIDØ +DPSIDØ
   PHIDØ =PHIDØ +DPHIDØ
   ALTØ  =(ALTØ +DALTØ)*FTTOM
   VEØK  =VEØK  +DVEØK
   XØ    =XØ    +DXØ
   YØ    =YØ    +DYØ
   GAMMAR=GAMMAR+DGAMMR

"*** DEFINE INITIAL Z COORDINATE. ***"

   ZØ =-ALTØ
```

```
        "*** ADD RUNTIME INCREMENTS TO PARAMETERS ***"

           VECIN(1)=ALTØ
           VECIN(2)=DRPM
           VECIN(3)=DXCGP
           VECIN(4)=DPL
           VECIN(5)=DEL5
           VECIN(6)=DEL6

        CALL PARINC(VECIN,CGPOS,PLS)

        "*** CONVERT FROM EAS TO TAS ***"

        PROCEDURAL (RHO=ALTØ)

        CALL ATMOS (ALTØ,RHO)

END     $" OF ATMOS PROCEDURAL"

           VTØK=VEØK*SQRT (RHOØ/RHO)

           VEØ = VEØK*KTOMPS
           VTØ = VTØK*KTOMPS


        "************ SPECIFY SYSTEM EXCITATION PARAMETERS ***********"


        "*** SPECIFY TERMINATION CONDITION ***"

        CONSTANT   TSTOP =Ø.Ø


        "*** SPECIFY INDEPENDENT VARIABLE AS A PRECAUTION ***"

        VARIABLE    TIME = Ø.Ø
        CINTERVAL   CINT = Ø.Ø
        NSTEPS      NSTP = Ø


        "*** SET INITIAL APPROXIMATION FOR INITIAL CONDITIONS ***"

           PSIRØ = PSIDØ*DEGTOR
           PHIRØ = PHIDØ*DEGTOR

        PROCEDURAL (PØ,QØ,RØ,BETARØ,ALPHRØ,THETRØ,ETARØ       ...
                   ,GAMMAR=VTØ,PHIRØ,G)

        CALL TRAP (VTØ,PHIRØ,G,GAMMAR,PØ,QØ,RØ,BETARØ         ...
                  ,ALPHRØ,THETRØ,ETARØ)

END     $" OF TRAP PROCEDURAL "


        "*** CALCULATE INERTIAL CONSTANTS ***"

           CØ  = ((IXX*IZZ)-(IXZ*IXZ))
           C1  = IZZ/CØ
           C2  = IXZ/CØ
           C3  = C2*(IXX-IYY+IZZ)
           C4  = ((IYY-IZZ)*C1)-(IXZ*C2)
           C5  = 1.Ø/IYY
           C6  = C5*IXZ
           C7  = C5*(IZZ-IXX)
           C8  = IXX/CØ
           C9  = ((IXX-IYY)*C8)+(IXZ*C2)
```

```
        "***************** INITIALIZE QUATERNIONS *******************"

       THETDØ= THETRØ*RADTOD

1Ø ..  CONTINUE

       "**NOTE:-THE EULER ANGLE(S) ARE HALVED WHEN CALCULATING...
              THE QUATERNIONS"

       CALL QUATNS(PSIRØ,THETRØ,PHIRØ,TAUØØ,TAU1Ø,TAU2Ø,TAU3Ø)

END $"OF INITIAL"

       "********************** DYNAMIC SECTION *********************"

DYNAMIC

       "*** THE TRANSITION FROM INITIAL TO DYNAMIC TRANSFERS    ***"
       "*** ALL INITIAL CONDITIONS TO THE STATE VARIABLES AND   ***"
       "*** EVALUATES THE CODE IN THE DERIVATIVE SECTION ONCE.  ***"

       IF(BEGIN) GO TO 2Ø

       "*** TRIM AIRCRAFT USING SUBROUTINE POWIT ***"

          XH(1) =THETRØ
          XH(2) =ALPHRØ
          XH(3) =ETARØ

       PROCEDURAL(XH=XH,MAXRES,ITLIM,ERRMAX)

       CALL POWIT(XH,MAXRES,ERRMAX,ITLIM)

          THETRØ =XH(1)
          ALPHRØ =XH(2)
          ETARØ  =XH(3)

END    $ "OF POWIT PROCEDURAL"

       BEGIN=.TRUE.
       GO TO 1Ø

2Ø ..  CONTINUE

       "******************** DERIVATIVE SECTION ********************"

DERIVATIVE

       "***************** CALCULATE CONTROL INPUTS *****************"

       PROCEDURAL(ETAR,PLS=                                  ...
       ETARØ,DELPL,ESTART,EREPET,EPULSE,TSTART,TREPET,TPULSE)
```

```
      DELETA=PULMAX*PULSE(ESTART,EREPET,EPULSE)
      DELPL =THRMAX*PULSE(TSTART,TREPET,TPULSE)
      ETAR=ETARØ+DELETA

   CALL CONTROLS(ETAR,DELPL,PLS)

      ETAD=ETAR*RADTOD

END    $ "OF CONTROLS PROCEDURAL"


   "********** CALCULATE QUATERNIONS; NORMALIZE, AND THEN **********"
   "**********        DETERMINE THE DIRECTION COSINES        **********"

   PROCEDURAL(L1,L2,L3,M1,M2,M3,N1,N2,N3=TAUØ,TAU1,TAU2,TAU3)

   "*** NORMALIZE QUATERNIONS ***"

      TAUN  = SQRT((TAUØ*TAUØ)+(TAU1*TAU1)+(TAU2*TAU2)+(TAU3*TAU3))
      TAUØN = TAUØ/TAUN
      TAU1N = TAU1/TAUN
      TAU2N = TAU2/TAUN
      TAU3N = TAU3/TAUN

   "*** CALCULATE THE DIRECTIONAL COSINES (L1 -> N3) ***"

      L1    = (((TAUØN*TAUØN)+(TAU1N*TAU1N))*2.Ø)-1.Ø
      L2    =  ((TAU1N*TAU2N)+(TAUØN*TAU3N))*2.Ø
      L3    =  ((TAU1N*TAU3N)-(TAUØN*TAU2N))*2.Ø

      M1    =  ((TAU1N*TAU2N)-(TAUØN*TAU3N))*2.Ø
      M2    = (((TAUØN*TAUØN)+(TAU2N*TAU2N))*2.Ø)-1.Ø
      M3    =  ((TAU2N*TAU3N)+(TAUØN*TAU1N))*2.Ø

      N1    =  ((TAU1N*TAU3N)+(TAUØN*TAU2N))*2.Ø
      N2    =  ((TAU2N*TAU3N)-(TAUØN*TAU1N))*2.Ø
      N3    = (((TAUØN*TAUØN)+(TAU3N*TAU3N))*2.Ø)-1.Ø

END $"OF QUATERNION PROCEDURAL"


   "*** CALCULATE ATMOSPHERIC CONDITIONS FOR CURRENT ALTITUDE ***"

   CALL ATMOS (-Z,RHO)


   "*** CALCULATE LONGITUDINAL AND LATERAL AERODYNAMIC FORCES ***"
   "***       IN STABILITY AXES AND MOMENTS IN BODY AXES        ***"

PROCEDURAL (XA,YA,ZA,LA,MA,NA=VT,ALPHAR,BETAR,P,Q,R,PHIR,THETAR,DALPHR)

   CALL AERO (XA,YA,ZA,LA,MA,NA,VT,ALPHAR,BETAR,P,Q,R,   ...
              PHIR,THETAR,DALPHR)

END $"OF AERO PROCEDURAL"


PROCEDURAL (XP,YP,ZP,LP,MP,NP=VT,ALPHAR,BETAR,P,Q,R,PHIR,THETAR)

   CALL PROP (XP,YP,ZP,LP,MP,NP,VT,ALPHAR,BETAR,P,Q,R,   ...
              PHIR,THETAR)

END $"OF PROP PROCEDURAL"
```

```
   SA = SIN(ALPHAR)
   CA = COS(ALPHAR)
   SB = SIN(BETAR)
   CB = COS(BETAR)

"*** CALCULATE GRAVITY FORCE COMPONENTS -> STABILITY AXES ***"

   FGX = L3*MASS*G*CA + N3*MASS*G*SA
   FGY = M3*MASS*G
   FGZ =-L3*MASS*G*SA + N3*MASS*G*CA

"*** CALCULATE THE TOTAL FORCES (FXS,FYS,FZS) ***"
"***              -> STABILITY AXES          ***"

   FXS = XA + XP*CA + ZP*SA + FGX
   FYS = YA + YP            + FGY
   FZS = ZA - XP*SA + ZP*CA + FGZ

"*** CALCULATE THE TOTAL MOMENTS (L,M,N) -> BODY AXES ***"

   L   = (LA*CA) - (NA*SA) + LP
   M   = MA                + MP
   N   = (LA*SA) + (NA*CA) + NP

"*** CALCULATE THE TOTAL FORCES (FXW,FYW,FZW) ***"
"***              -> AIR-PATH AXES           ***"

   FXW = FXS*CB + FYS*SB
   FYW =-FXS*SB + FYS*CB
   FZW = FZS

"*** CALCULATE FORCES IN BODY AXES ***"

   FXB =  XA*CA - ZA*SA + XP
   FYB =  YA + YP
   FZB =  XA*SA + ZA*CA + ZP

"*** CALCULATE LINEAR ACCELERATIONS ***"

   AX  =  FXB/(MASS*G)
   AY  =  FYB/(MASS*G)
   AZ  =  FZB/(MASS*G)

"*** CALCULATE NORMAL ACCELERATION ***"

   AN  = -AZ

"*** RESOLVE BODY AXIS ANGULAR RATES INTO STABILITY AXES ***"

   PS  =  P*CA + R*SA
   RS  = -P*SA + R*CA

"****** CALCULATE LINEAR DERIVATIVES -> AIR PATH AXES ******"

"*** DECLARE DALPHR AS AN IMPLICIT VARIABLE ***"

   DALPHR = IMPL (Q,Ø.ØØ1,3Ø,EF,          ...
```

```
                    (Q*CB-PS*SB+FZW/MASS/VT)/CB,Ø.ØØØ1)

PROCEDURAL (=EF)

      CALL IMP(EF)

END $"OF DALPHR PROCEDURAL"


         DBETAR = FYW/MASS/VT-RS
         DVT    = FXW/MASS


        "*** CALCULATE ANGULAR ACCELERATIONS -> BODY AXES ***"

         PDOT =  (L*C1)+(N*C2)+(((P*C3)+(R*C4))*Q)
         QDOT =  (M*C5)+(((R*R)-(P*P))*C6)+(R*P*C7)
         RDOT =  (N*C8)+(L*C2)+(((P*C9)-(R*C3))*Q)



       "********* DETERMINE QUATERNION RATES AND EULER ANGLES. *********"


       "*** CALCULATE THE QUATERNION RATES ***"

         TAUØDT=-(((TAU1N*P)+(TAU2N*Q)+(TAU3N*R))*Ø.5)
         TAU1DT=  ((TAUØN*P)-(TAU3N*Q)+(TAU2N*R))*Ø.5
         TAU2DT=  ((TAU3N*P)+(TAUØN*Q)-(TAU1N*R))*Ø.5
         TAU3DT=-(((TAU2N*P)-(TAU1N*Q)-(TAUØN*R))*Ø.5)


       "*** CALCULATE THE NEW EULER ANGLES ***"

PROCEDURAL(PSID,THETAD,PHID=L1,L2,L3,M3,N3)


       "*** CALCULATE HEADING (OR YAW) ANGLE (PSI)  ***"
       "***      IN RANGE Ø(NORTH) TO 36Ø DEG.       ***"

         PSIR = ATAN(L2/L1)
         IF(PSIR.LT.Ø.Ø) PSIR = PSIR+PI*2
         PSID = PSIR*RADTOD


       "*** CALCULATE ANGLE OF PITCH (THETA) ***"
       "***      IN RANGE +/-9Ø DEGREES       ***"

         THETAR= ATAN(-L3/SQRT((L1*L1)+(L2*L2)))
         THETAD= THETAR*RADTOD


       "***   CALCULATE BANK (OR ROLL) ANGLE (PHI) IN RANGE    ***"
       "*** +/-18Ø DEGREES,WHERE Ø DEG. INDICATES WINGS LEVEL ***"

         PHIR = ATAN(M3/N3)
         PHID = PHIR*RADTOD

END $"OF EULER PROCEDURAL"

        "***********************  TRAJECTORY  **********************"


        "***     RESOLVE FLIGHT PATH AXES VELOCITIES INTO BODY     ***"
        "*** COMPONENTS FOR CALCULATION OF DISTANCES IN EARTH AXES ***"
```

```
          UB = VT*CA*CB
          VB = VT*SB
          WB = VT*SA*CB


       "***      CALCULATE EARTH AXES VELOCITIES FROM BODY AXES      ***"
       "*** VELOCITIES USING DIRECTION COSINES AND WIND VELOCITIES ***"

          VNE = L1*UB+M1*VB+N1*WB-VWN
          VEE = L2*UB+M2*VB+N2*WB-VWE
          VDE = L3*UB+M3*VB+N3*WB-VWD


PROCEDURAL(VGRKT,GAMMAD,CHID=VNE,VEE,VDE)


       "*** CALCULATE THE VELOCITY OVER THE GROUND, ***"
       "***          VGR (IE. GROUND SPEED)         ***"

          VGR   = SQRT((VNE*VNE)+(VEE*VEE))
          VGRKT = VGR*MPSTOK


      "*** CALCULATE FLIGHT-PATH ANGLE, GAMMA IN RANGE +/-9Ø DEG ***"

         GAMMAR = THETAR-ALPHAR
         GAMMAD = GAMMAR*RADTOD


      "*** CALCULATE ANGLE OF CLIMB, LAMBDA IN RANGE +/-9Ø DEG ***"

         LAMDAR = ATAN(-VDE/VGR)
         LAMDAD = LAMDAR*RADTOD


      "*** CALCULATE ANGLE OF TRACK, CHI, ***"
      "*** IN RANGE Ø(NORTH) TO 36Ø DEG.  ***"

         CHIR   = ATAN(VEE/VNE)
         CHID   = CHIR*RADTOD

END $"OF TRAJECTORY PROCEDURAL"


      "*** CALCULATE THE RANGE (NOTE: IN METRES) ***"

         RANGE  = SQRT((X*X)+(Y*Y))


      "*********** INTEGRATION OF SYSTEM STATE EQUATIONS ***********"

         ALPHAR = INTVC(DALPHR,ALPHRØ)
         BETAR  = INTVC(DBETAR,BETARØ)
         VT     = INTVC(DVT,VTØ)

         P      = INTVC(PDOT,PØ)            $"ROLL RATE"
         Q      = INTVC(QDOT,QØ)            $"PITCH RATE"
         R      = INTVC(RDOT,RØ)            $"YAW RATE"

         TAUØ   = INTVC(TAUØDT,TAUØØ)       $"QUATERNION TERMS"
         TAU1   = INTVC(TAU1DT,TAU1Ø)
         TAU2   = INTVC(TAU2DT,TAU2Ø)
         TAU3   = INTVC(TAU3DT,TAU3Ø)

         Z      = INTEG(VDE,ZØ)             $"Z-POSITIONAL CO-ORDINATE"
         X      = INTEG(VEE,XØ)             $"X-POSITIONAL CO-ORDINATE"
```

```
        Y      = INTEG(VNE,YØ)           $"Y-POSITIONAL CO-ORDINATE"

END $"OF DERIVATIVE"

        VE     = VT*SQRT(RHO/RHOØ)       $"EQUIVALENT AIRSPEED  (M/S)"
        VEK    = VE*MPSTOK               $"EQUIVALENT AIRSPEED  (KTS)"
        ALPHAD = ALPHAR*RADTOD           $"ANGLE OF ATTACK      (DEG)"
        ALT    =-Z/FTTOM                 $"ALTITUDE             (FT.)"

     "**************** EXPRESS TERMINATION CONDITION ****************"

     TERMT(TIME.GE.TSTOP)

END $"OF DYNAMIC"

     "*** NOTE:   THIS LISTING SHOULD BE USED IN CONJUNCTION WITH ***"
     "*** THE ADVANCED CONTINUOUS SIMULATION LANGUAGE (ACSL) USER ***"
     "*** GUIDE/REFERENCE MANUAL.                                 ***"

END $"OF PROGRAM"

      SUBROUTINE QUATNS(PSIR,THETAR,PHIR,TAUØ,TAU1,TAU2,TAU3)
C     "*** CALCULATES NORMALIZED QUATERNIONS ***"

         SPSI   = SIN(PSIR*Ø.5)
         CPSI   = COS(PSIR*Ø.5)
         STHETA = SIN(THETAR*Ø.5)
         CTHETA = COS(THETAR*Ø.5)
         SPHI   = SIN(PHIR*Ø.5)
         CPHI   = COS(PHIR*Ø.5)

C     "*** CALCULATE THE QUATERNIONS FROM THE EULER ANGLE TERMS ***"

         TAUØ = (CPHI*CTHETA*CPSI)+(SPHI*STHETA*SPSI)
         TAU1 = (SPHI*CTHETA*CPSI)-(CPHI*STHETA*SPSI)
         TAU2 = (CPHI*STHETA*CPSI)+(SPHI*CTHETA*SPSI)
         TAU3 = (CPHI*CTHETA*SPSI)-(SPHI*STHETA*CPSI)

C     "*** NORMALIZE INITIAL QUATERNIONS ***"

         TAUN = SQRT((TAUØ*TAUØ)+(TAU1*TAU1)+(TAU2*TAU2)
     +          +(TAU3*TAU3))
         TAUØ = TAUØ/TAUN
         TAU1 = TAU1/TAUN
         TAU2 = TAU2/TAUN
         TAU3 = TAU3/TAUN

      RETURN

      END
```

```
      SUBROUTINE EVAL (XXXX,XXXRES,MAXR)

C     "*** THIS SUBROUTINE IS CALLED FROM THE TRIMMING SUBROUTINE ***"
C     "***          TO GAIN ACCESS TO THE DERIVATIVE SECTION.      ***"

      DIMENSION XXXX(3),XXXRES(3)
$
C               "*** THE "$" CONTROL CHARACTER MAKES MAIN PROGRAM ***"
C               "***    VARIABLES AVAILABLE TO THIS SUBROUTINE.   ***"

      INTEGER I
      I=1

      THETAR = (XXXX(1))
      ALPHAR = (XXXX(2))
      ETARØ  = (XXXX(3))

      CALL QUATNS(PSIRØ,THETAR,PHIRØ,TAUØ,TAU1,TAU2,TAU3)

      CALL ZZDERV(I)

      XXXRES(1) = (DVT)
      XXXRES(2) = (DALPHR)
      XXXRES(3) = (QDOT)

      RETURN
      END



      SUBROUTINE IMP(EF)

C     "***** NOTIFIES FAILURE OF IMPLICIT ITERATION ROUTINE *****"


      IF (EF .EQ. Ø.Ø) GOTO 89
      WRITE (6,99)
99    FORMAT(//27HIMPLICIT PROCEDURAL FAILED   //)

89    RETURN
      END
```

FTSUBS.F

```
      SUBROUTINE CAFCDI(PHIN,HNM,VN,IXX,IYY,IZZ,IXZ,MASS)

      IMPLICIT REAL(A-Z)
      INCLUDE 'FTPAR.F'
      INTEGER I,J

C           ROUTINE FOR THE INPUT OF CONFIGURATION AND
C                      FLIGHT CONDITION DATA.

      OPEN  (UNIT=1,FILE='FTSUD2.IN',STATUS='OLD')
      OPEN  (UNIT=7,FILE='FTGRAF.VPP',STATUS='OLD')


C     *** READ IN CONFIGURATION AND FLIGHT CONDITION DATA ***


      READ(1,*)TTOT,TSAMP,NH,HN,DELH,NV,VN,DELV,NPHI,PHIN
      READ(1,*)DELPHI,NPL,PLØ,DELPL,NRPM,RPM,DELRPM,WEGHT,NXCG,XCGP
      READ(1,*)DELXCG,ZCG,(PEFF(I),I=1,7),XT
      READ(1,*)ZT,XTH,ZTH,THSET,MAXEP,NBLAD,PDIA,WSET,TPSET,ETAG
      READ(1,*)ACMASS,ACIXX,ACIYY,ACIZZ,ACIXZ,SW,CW,BW,ST,CT
      READ(1,*)CETA,XP,ZP,CWP,BFW,BTAIL,CLØ,CLAL,CDØ,CDAL
      READ(1,*)CMØ,CMAL,EPSØ,EPSAL,QTOQ,AØ,A1,A2,A3,BØ
      READ(1,*)B1,B2,B3,CDØT,CDLT,CMTØ,CMQW,CLP,CLXI,KWING
      READ(1,*)KFUSE,NPSFI,NPSFR,MPSFI,MPSFR,CDB,GTR,BLØ,WTR,TAPERF
      READ(1,*)XWC,ZWC,SØS,XQARTC

      READ(7,*) (FPROP(I,1),FPROP(I,2),I=1,2Ø)
      READ(7,*) ((TOP(I,J),J=1,33),I=1,2Ø)
      READ(7,*) ((ETP(I,J),J=1,6Ø),I=1,21)
111   READ(7,*) (CYPROP(I,1),CYPROP(I,2),CYPROP(I,3),CYPROP(I,4),I=1,2Ø)
      READ(7,*) ((DELEPS(I,J),J=1,1Ø),I=1,8)



      MASS=ACMASS
      IXX=ACIXX
      IYY=ACIYY
      IZZ=ACIZZ
      IXZ=ACIXZ

      HNM=HN

      CLOSE(UNIT=1)
      CLOSE(UNIT=7)


      RETURN

      END


      SUBROUTINE PARINC (VECIN,CGPOS,PLS)

C     " SUBROUTINE TO ALLOW RUNTIME PARAMETER MODIFICATIONS "

      IMPLICIT REAL(A-Z)
      DIMENSION VECIN(6)
      INCLUDE 'FTPAR.F'
```

```
      ALTØ=      VECIN(1)
      RPM = RPM+VECIN(2)
      XCGP=XCGP+VECIN(3)
      PLØ =PLØ +VECIN(4)
      PAR5=PAR5+VECIN(5)
      PAR6=PAR6+VECIN(6)


      XCG=XCGP*CW/1ØØ.+XQARTC-CW/4.
      CGPOS=XCGP
      PLS=PLØ

      RETURN

      END


      SUBROUTINE TRAP(VTØ,PHIRØ,G,GAMMAR,PØ,QØ,RØ,BETARØ
     1                   ,ALPHRØ,THETRØ,ETARØ)

C         " GIVES INITIAL APPROXIMATION FOR TRIM CONDITION "

      IMPLICIT REAL(A-Z)
      INCLUDE 'FTPAR.F'

      PI = 3.141593
      ETAØØ= 2.2
      KK =41.Ø*(XCGP/1ØØ.-Ø.4)
      QD =(.5*1.2256*VTØ*VTØ)

      PØ =Ø.
      QØ =G/VTØ*TAN(PHIRØ)*SIN(PHIRØ)
      RØ =G/VTØ*SIN(PHIRØ)

      AN  = 1/COS(PHIRØ)

      BETARØ = Ø.


      ALPHAW = ACMASS*G/(CLAL*QD*SW)
      ALPHRØ = ALPHAW - WSET
      CLWB   = CLØ + (CLAL*ALPHAW) + CLF
      CDWB   = CDØ + (CDAL*CLWB**2) + CDF

      DRAG   = CDWB * QD * SW

      GAMMAR = ASIN(-DRAG/(ACMASS*G))
      THETRØ = GAMMAR + ALPHRØ

      ETAØ   = (ETAØØ+KK*CLWB)*PI/18Ø.
      ETARØ  = ETAØ




      RETURN

      END

      SUBROUTINE ATMOS  (H,RO)

      IMPLICIT REAL(A-Z)
```

```
      COMMON/ARGS2/HITE,DENS

      HITE=H
      DENS=1.2256*(1-Ø.Ø2256*HITE/1ØØØ.Ø)**4.2561
      RO  =DENS


      RETURN

      END


      SUBROUTINE AERO (XA,YA,ZA,LA,MA,NA,VT,ALPHAR,BETAR
     1                ,PP,QQ,RR,PHIR,THETAR,DALPHR)
C
C MODIFIED BY RODD PERRIN OCTOBER 1985
C

      IMPLICIT REAL(A-Z)
      INCLUDE 'FTPAR.F'
      DIMENSION X(2Ø)

      XDAL = DALPHR

      X(V) = VT
      X(ALPHA) = ALPHAR
      X(Q) = QQ
      X(P) = PP
      X(H) = HITE
      X(PHI) = PHIR
      X(THETA) = THETAR

      CALL ATMOS(HITE,RHO)

      QD=.5*RHO*VT**2

      CALL THRUST(X,TOP,ETP)
      CALL PROPEL(X)
      CALL FLAPS(X)
      CALL WB(X)
      CALL TAIL(X)
      CDW=CDWB-CDB
      CALL RESOLV(X(ALPHA),CLWB,CDW,XP,ZP,CMWBF)
      ALTF=ALPHAT-TPB
      CALL RESOLV(ALTF,CLT,CDT,XT,ZT,CMTF)


      CTHXW=CTPROP*COS(ALTH)-CLPROP*SIN(ALTH)
      CTHZW=-CLPROP*COS(ALTH)-CTPROP*SIN(ALTH)

      CTHX=CTPROP*COS(THSET)-CLPROP*SIN(THSET)
      CTHZ=-CLPROP*COS(THSET)-CTPROP*SIN(THSET)

      ALTFS=X(ALPHA)-ALTF
      CTXW=-CLT*SIN(ALTFS)-CDT*COS(ALTFS)
      CTZW=-CLT*COS(ALTFS)+CDT*SIN(ALTFS)

      CMTHF=CTHX*(ZTH-ZCG)+CTHZ*(XTH-XCG)
      CLWBT=CLWB-ST*(CTZW)/SW

      CALL RESOLV(X(ALPHA),Ø.Ø,CDB,XP,ZTH,CMBF)
      CMWBTH=CMWB+(CMWBF+CMTHF+CMBF)/CW

C     ***  CALCULATE AERODYNAMIC FORCES  ***

      XA  = QD*(SW*(CTHXW CDWP) ST*CTXW)
      YA  = Ø.Ø
```

```
      ZA  = QD*(SW*(CTHZW-CLWBT))

      LA = Ø.Ø
      MA = QD*(SW*CW*CMWBTH+(ST*(CT*CMT+CMTF))+(SW*CW*CMQW*Q))
      NA = Ø.Ø


      RETURN
      END



      SUBROUTINE PROP (XP,YP,ZP,LP,MP,NP,VT,ALPHAR,BETAR
     1                 ,PP,QQ,RR,PHIR,THETAR)


      RETURN

      END


      SUBROUTINE CONTROLS(ETAR,DELPL,PLS)

      INCLUDE 'FTPAR.F'

      ETA = ETAR
      PL  = PLØ+DELPL
      PLS = PL

      RETURN

      END
```

FTSUBS2.F

```
      SUBROUTINE THRUST(X)
C     !CALCULATES COMPONENTS OF PROPELLOR THRUST
      IMPLICIT REAL (A-Z)
      INCLUDE 'FTPAR.F'
      INTEGER I,J
      DIMENSION X(2Ø)

1ØØØ  FORMAT(' POWER TOO LOW')
1ØØ1  FORMAT(' POWER TOO HIGH')

4     EP=PL*MAXEP
      ETAP=Ø
      TCC=Ø.Ø
      TTHST=Ø.Ø
      IF(RPM==Ø)GO TO 1
      IF(PL==Ø)GO TO 1

      IF(X(V)>55.Ø)GO TO 5

      POD2=SØS*EP/745.7/((PDIA/Ø.3Ø48)**2)
      PIND=3.142*RPM*PDIA/Ø.3Ø48/6Ø.Ø

      DO 3 J=1,6
      DO 2 I=1,5
2     CALL INTERP(TOP(1,1),TOP(1,(J-1)*5+I+3),2Ø,POD2,TOP(I,34),ERROR)
3     CALL INTERP(TOP(1,2),TOP(1,34),5,PIND,TOP(J,35),ERROR)
      CALL INTERP(TOP(1,3),TOP(1,35),6,X(V),TDP,ERROR)

      IF(TDP<Ø)GO TO 9
      IF(TDP>1Ø)GO TO 1Ø
      TTHST=TDP*EP*4.4497/745.7
      TCC=TTHST/(2.Ø*QD*PDIA**2)
      ETAP=TTHST*X(V)/EP
      RETURN

5     SØSCP=SØS*EP/DENS/((RPM/6Ø)**3)/((PDIA)**5)
      IF(SØSCP<Ø.Ø25)GO TO 9

      MACH=X(V)/34Ø.29
      PINDOA=3.142*RPM*PDIA/6Ø.Ø/34Ø.29

      DO 8 J=1,5
      DO 7 I=1,11

7     CALL INTERP(ETP(1,1),ETP(1,(J-1)*11+I+3),21,SØSCP,ETP(I,59)
     1 ,ERROR)
8     CALL INTERP(ETP(1,2),ETP(1,59),11,PINDOA,ETP(J,6Ø),ERROR)
      CALL INTERP(ETP(1,3),ETP(1,6Ø),5,MACH,ETAP,ERROR)

      IF(ETAP>1.Ø)GO TO 9
      IF(ETAP<Ø.Ø)GO TO 11
1     TTHST=ETAP*EP/X(V)
      TCC=TTHST/(2.Ø*QD*PDIA**2)
      RETURN
9     IF(RPM<6ØØ)GO TO 12
      RPM=RPM-2ØØ
      GO TO 4
12    TTHST=-1.Ø
      TYPE 1ØØØ
      RETURN
1Ø    IF(RPM>26ØØ)GO TO 13
```

```
      RPM=RPM+2ØØ
      GO TO 4
13    ETAP=-1.Ø
      TYPE 1ØØ1
      RETURN
11    IF(RPM<6ØØ)GO TO 14
      RPM=RPM-2ØØ
      GO TO 4
14    TCC=-1.Ø
      TYPE 1ØØ1
      RETURN
      END

      SUBROUTINE WB(X)
C     !CALCULATES WING-BODY AERODYNAMICS
      IMPLICIT REAL(A-Z)
      INCLUDE 'FTPAR.F'
      DIMENSION X(2Ø)

      CLWB=CLØ+(CLAL*ALPHAW)+CLF
      CDWB=CDØ+(CDAL*(CLWB**2))+CDF
      CMWB=CMØ+(CMAL*ALPHAW)+CMF

      CLSS=Ø
      CMSS=Ø
      QSOQIN=TCC*8.Ø/3.142
      TEMP=DUWDAL

      CALL SLIPST(X,2)
      IF(ZSS>(PDIA/2.Ø))GO TO 2
      BWP=2*SQRT((PDIA/2.Ø)**2-ZSS**2)-BFW/2
      IF(PEFF(2)==Ø)GO TO 1
      CMSS=(CMØ+CMF)*CWP**2*BWP*QSOQIN/CW/SW
1     IF(PEFF(3)==Ø)GO TO 2

      DELAW=-(DPEDAL*ALTH)/(1+TEMP)
      AIL=BWP/CWP
      SIL=BWP*CWP
      BL=Ø.Ø8
      ARL=BW/CW
      IF(ARL>1Ø.Ø)BL=Ø.Ø
      K2=3.Ø*(AIL-1.Ø)+BL*(1Ø.Ø-ARL)*AIL
      K3=1.96*SQRT(TCC)
      K4=(K2+3.Ø)*Ø.1*K3
      K1=Ø.2*K4+1.181-Ø.489*SQRT(TCC+Ø.1)
      CLSS=K1*SIL*((1+QSOQIN)*CLAL*DELAW+QSOQIN*CLWB)/SW

2     CLWB=CLWB+CLSS
      CMWB=CMWB+CMSS

      CDWB=CDWB*(1.Ø+CLSS/CLWB)

      XZSS(18)=ZWC
      XZSS(19)=ZT
      XZSS(2Ø)=PDIA

      RETURN
      END


      SUBROUTINE TAIL(X)
C     !CALCULATES TAIL CONTRIBUTION
      IMPLICIT REAL(A-Z)
      INCLUDE 'FTPAR.F'
      DIMENSION X(2Ø)

      CALL DWASH(X)
```

```
      CALL BEND(X)


      CALL SLIPST(X,3)
      IF(ZHEFF>(PDIA/2.Ø))GO TO 1
      SHI=2*CT*SQRT((PDIA/2.Ø)**2-ZHEFF**2)
      K1=4.Ø8*SHI/ST
      K2=K1*(TCC+Ø.575)-Ø.3
      K3=Ø.2Ø2-Ø.Ø3*ZHEFF/(PDIA/2.Ø)-Ø.Ø754*((ZHEFF/(PDIA/2.Ø))**2)
      DELQT=K2*K3-Ø.1
      IF(TCC<Ø.1)DELQT=Ø.Ø
      GO TO 2

1     DELQT=Ø
2     RT=QTOQ+DELQT
      RT1=RT-1
      IF(PEFF(5)==Ø)RT=1.Ø

      ALPHAQ=(XT-XCG)*X(Q)/(X(V)*SQRT(RT))
      ALPHAT=X(ALPHA)+TPB-EPS+ALPHAQ
      CLT=RT*(AØ+A1*ALPHAT)+A2*ETA+A3*BETA
      CH=RT*(BØ+B1*ALPHAT)+B2*ETA+B3*BETA

      IF (PEFF(6)==Ø)RT1=Ø.Ø

      CLT=CLT+RT1*(A2*ETA+A3*BETA)
      CH=CH+RT1*(B2*ETA+B3*BETA)
      CDT=CDØT+CDLT*CLT**2

      CLTTH=CLT-RT*(A2*ETA+A3*BETA)
      CDTTH=CDØT+CDLT*CLTTH**2

      CMT=CMTØ
      PETA=QD*CETA*CH*ETAG

      XZSS(22)=CT

      RETURN
      END


      SUBROUTINE RESOLV(AL,CL1,CD1,X1,Z1,CM1)
C     !TRANSLATES FORCES TO THE CG
      IMPLICIT REAL(A-Z)
      INCLUDE 'FTPAR.F'

      CM1=-(CD1*COS(AL)-CL1*SIN(AL))*(Z1-ZCG)
     1 -(CL1*COS(AL)+CD1*SIN(AL))*(X1-XCG)

      RETURN
      END


      SUBROUTINE DWASH(X)
C     !CALCULATES DOWNWASH AND TOTAL HEAD AT THE TAIL
C     !AND ALSO INDUCED INCIDENCE AT THE TAIL DUE TO PITCHING
      IMPLICIT REAL(A-Z)
      INTEGER I
      INCLUDE 'FTPAR.F'
      DIMENSION X(2Ø)

      KA=CW/BW-1.Ø/(1.Ø+(BW/CW)**1.7)
      KLAM=(1Ø.Ø-3.Ø*WTR)/7.Ø
      LH=XT-XWC
      HH=(XT-XWC)*TAN(WSET)+ZWC-ZT
      KH=(1-HH/BW)/((2.Ø*LH/BW)**Ø.33)
```

```
      EPSAL=4.44*((KA*KLAM*KH)**1.19)

      EPS=57.3*(EPSØ+EPSAL*(ALPHAW-XDAL*(XT-XCG)/X(V)))

      IF(PEFF(4)==Ø)GO TO 1
      DO 2 I=1,8
2     CALL INTERP(DELEPS(1,1),DELEPS(1,I+2),8,EPS,DELEPS(I,11),ERROR)
      CALL INTERP(DELEPS(1,2),DELEPS(1,11),8,TCC,DEP,ERROR)

      ZHT=ZTH+(XT-XTH)*TAN(THSET)-ZT
      KFAC=Ø.85-Ø.25*ABS(ZHT)/PDIA
      DEPS=KFAC*DEP
      EPS=EPS+DEPS

1     EPS=EPSF+(EPS)/57.3
      QTOQ=QTOQ
      ALPHAQ=(XT-XCG)*X(Q)/X(V)

      XZSS(21)=CW

      RETURN
      END


      SUBROUTINE BEND(X)
C     !CALCULATES TAILPLANE ANGLE DUE TO FUSELAGE BENDING
      IMPLICIT REAL(A-Z)
      INCLUDE 'FTPAR.F'
      DIMENSION X(2Ø)

      LT=QD*QTOQ*ST*CLT
      TPB=KFUSE*LT+TPSET

      RETURN
      END


      SUBROUTINE TORSON(X)
C     !CALCULATES EFFECTIVE WING INCIDENCE
      IMPLICIT REAL(A-Z)
      INCLUDE 'FTPAR.F'
      DIMENSION X(2Ø)

      LW=QD*SW*CLWB
      ALPHAW=X(ALPHA)+WSET-KWING*LW

      RETURN
      END


      SUBROUTINE PROPEL(X)

C     !CALCULATES PROPELLER FORCES

      IMPLICIT REAL(A-Z)
      INCLUDE 'FTPAR.F'
      DIMENSION X(2Ø)
      INTEGER IB
      CALL TORSON(X)
      CALL SLIPST(X,1)
      ALTH=THSET+X(ALPHA)+DUWDAL*ALPHAW
      IF(RPM==Ø)GO TO 1

      DPEDAL=Ø
```

```
      CLPROP=Ø
      IF(PEFF(1)==Ø)GO TO 3
      IB=IFIX(NBLAD)
      CALL INTERP(CYPROP(1,1),CYPROP(1,IB),2Ø,BLØ,CYPSIØ,ERROR)

      CALL INTERP(FPROP(1,1),FPROP(1,2),2Ø,TCC,FPR,ERROR)
      CLPROP=3.142*PDIA**2*CYPSIØ*FPR*ALTH/SW/4.Ø
      C1=SQRT((SQRT(5Ø.6*TCC+28.623)-5.35)/25.3)
      C2=Ø.27*EXP(-Ø.127*TCC)
      DPEDAL=C1+C2*CYPSIØ
3     CTPROP=2*TCC*PDIA**2/SW

      GO TO 2
1     DPEDAL=Ø
      CLPROP=Ø
      CTPROP=Ø
2     CONTINUE

      XZSS(17)=ZTH

      RETURN
      END


      SUBROUTINE INTERP(XLIST,YLIST,N,X,Y,ERROR)
C USED FOR INTERPOLATING INTO A ONE DIMENSIONAL ARRAY
C     X:            INDEPENDENT VARIABLE
C XLIST:      LIST OF INDEPENDENT VARIABLE BREAK POINTS IN
C            ASCENDING ORDER
C YLIST:      LIST OF DEPENDENT VARIABLE BREAK POINTS
C     N:            NUMBER OF BREAK POINTS
C     Y:            INTERPOLATED VALUE OF DEPENDENT VARIABLE
C ERROR:      .TRUE. IF EXTRAPOLARION WAS NEEDED
C
      DIMENSION XLIST(1ØØ),YLIST(1ØØ)
      IF(X.GE.XLIST(1))GO TO 5
      ERROR=.TRUE.
      I=1
      GO TO 2Ø
5     IF(X.LE.XLIST(N))GO TO 7
      ERROR=.TRUE.
      I=N-1
      GO TO 2Ø
7     ERROR=.FALSE.
      DO 1Ø I=1,N-1
      IF(X.GE.XLIST(I).AND.X.LE.XLIST(I+1))GO TO 2Ø
1Ø    CONTINUE
      I=N-1
2Ø    J=I
      XLB=XLIST(J)
      XUB=XLIST(J+1)
      RX=XUB-XLB
      W1=(XUB-X)/RX
      W2=(X-XLB)/RX
      Y=W1*YLIST(J)+W2*YLIST(J+1)
      RETURN
      END


      SUBROUTINE SLIPST(X,I)

C     !CALCULATES LOCUS OF PROPELLER SLIPSTREAM


      IMPLICIT REAL(A-Z)
      INCLUDE 'FTPAR.F'
```

```
      DIMENSION X(2Ø)
      INTEGER I

      GO TO(1Ø,2Ø,3Ø),I

1Ø    XZSS(3)=XTH
      DUWDAL=Ø.28/(XWC-XTH)/CW
      XZSS(1)=2*XZSS(3)
      ALUW=X(ALPHA)+DUWDAL*ALPHAW
      IF(PEFF(7))12,12,11
11    ALUW=X(ALPHA)+DUWDAL*(CLWB/CLAL)
12    XZSS(2)=ABS(XZSS(3))*TAN(ALUW)+ZTH
      XZSS(4)=ZTH

      RETURN


2Ø    XZSS(5)=XZSS(3)/2
      ALUW=ALUW-DPEDAL*ALTH
      XZSS(6)=-ABS(XZSS(5))*TAN(ALUW)+XZSS(4)
      DUWDAL=Ø.28/(XWC-(XTH/2))/CW
      IF(PEFF(7))21,21,22
21    ALUW=ALUW+DUWDAL*ALPHAW
      GO TO 23
22    ALUW=ALUW+DUWDAL*(CLWB/CLAL)
23    XZSS(7)=XWC
      XZSS(8)=-ABS(XZSS(5)-XZSS(7))*TAN(ALUW)+XZSS(6)
      ZSS=ABS(ZWC-XZSS(8))

      ALUW=-WSET
      XZSS(9)=XWC+Ø.75*CW
      XZSS(1Ø)=-Ø.75*CW*TAN(ALUW)+XZSS(8)+ZFLAP

      RETURN


3Ø    KA=CW/BW-1/(1+(BW/CW)**1.7)
      KLAM=(1Ø.Ø-3.Ø*WTR)/7.Ø
      WTE=XWC+Ø.75*CW

      XZSS(11)=WTE+(XT-WTE)/2
      LH=XZSS(11)-XWC
      HH=(XZSS(11)-XWC)*TAN(WSET)+ZWC-XZSS(1Ø)
     1 -(XZSS(11)-WTE)*TAN(ALPHAW)
      KH=(1-HH/BW)/((2*LH/BW)**Ø.33)
      DDWDAL=4.44*((KA*KLAM*KH)**1.19)
      XZSS(12)=-ABS(XZSS(11)-WTE)*TAN(X(ALPHA)-(DDWDAL*ALPHAW)-EPSF)
     1 +XZSS(1Ø)

      XZSS(13)=XT
      LH=XT-XWC
      HH=(XT-XWC)*TAN(WSET)+ZWC-XZSS(12)
     1 -(XT-XZSS(11))*TAN(ALUW)
      KH=(1-HH/BW)/((2*LH/BW)**Ø.33)
      DDWDAL=4.44*((KA*KLAM*KH)**1.19)
      XZSS(14)=-ABS(XT-XZSS(11))*TAN(X(ALPHA)-(DDWDAL*ALPHAW)-EPSF)
     1 +XZSS(12)

      ZHEFF=ABS(ZT-XZSS(14))

      XZSS(15)=XT+(XT-WTE)/2
      LH=XZSS(15)-XWC
      HH=(XZSS(15)-XWC)*TAN(WSET)+ZWC
     1 -XZSS(14)-(XZSS(15)-XT)*TAN(ALUW)
      KH=(1-HH/BW)/((2*LH/BW)**Ø.33)
```

```
      DDWDAL=4.44*((KA*KLAM*KH)**1.19)
      XZSS(16)=-ABS(XZSS(15)-XT)*TAN(X(ALPHA)-(DDWDAL*ALPHAW)-EPSF)
     1 +XZSS(14)

      RETURN
      END
      SUBROUTINE FLAPS(X)

C     !CALCULATES CL,CM,CD,EPS,ZFLAP FOR A SINGLE SLOTTED
C     !FLAP FITTED TO THE A1.DEFLECTIONS 2Ø DEG AND 4Ø DEG

      IMPLICIT REAL(A-Z)
      INCLUDE 'FTPAR.F'

      CLF=Ø.Ø
      CMF=Ø.Ø
      CDF=Ø.Ø
      ZFLAP=Ø.Ø
      IF(PEFF(7)-1)15,5,1Ø

5     CLF=Ø.74
      CMF=-Ø.167
      CDF=Ø.Ø213
      ZFLAP=Ø.2*CW*Ø.364
      GO TO 15

1Ø    CLF=1.13
      CMF=-Ø.2829
      CDF=Ø.Ø628
      ZFLAP=Ø.2*CW*Ø.839

15    EPSF=(CLF*(1Ø.3+64.1*((ABS(ZWC-ZT)/(BW/2.Ø))-Ø.53)**2)/
     1 (2.Ø*BW*5.Ø/SW))/57.3
      XFLAP=Ø.2*CW

      RETURN
      END
```

FTPAR.F

```
      PARAMETER  (V=1,ALPHA=2,Q=3,P=4,H=5,PHI=6,THETA=7)

      INTEGER V,ALPHA,Q,P,H,PHI,THETA
      COMMON/ARGS/ACMASS,ACIXX,ACIYY,ACIZZ,ACIXZ,SW,CW,BW,WSET
     1 ,ST,CT,CETA,TPSET,ETAG,XP,ZP,XT,ZT,XQARTC
     1 ,XTH,ZTH,ZSS,XCG,XCGP,ZCG,AØ,A1,A2,A3,BØ,B1,B2,B3
     2 ,CLØ,CDØ,CMØ,CLAL,CDAL,CMAL,MAXEP,THSET,ALTH
     3 ,DUWDAL,ALUW,EPSØ,EPSAL,QTOQ,KFUSE,KWING,CDØT,CDLT,CMTØ
     4 ,CMQW,CLP,CLXI,TSAMP,VØ,PHIØ
     5 ,PW,QW,RW,CTHX,CTHZ,CLWB,CDWB,CMWB,CLWBT
     6 ,ALPHAW,TPB,ALPHAQ,EPS,DPEDAL,ALPHAT,CLT,CDT,CMT,CH
     7 ,ETAØ,ETA,PETA,BETA,XI
     8 ,TTHST,CTHXW,CTHZW,CMWBF,CMTF,CLTTH,CDTTH,CMWBTH,CMTTH
     9 ,NPSFI,NPSFR,MPSFI,MPSFR,CDB,GTR,QD,RHO,XDV,XDAL
     1 ,NBLAD,PDIA,RPM,CWP,BFW,CLPROP,CTPROP,ETAP,TCC,PCP
     2 ,WTR,TAPERF,BTAIL,DTAIL,BLØ,SØS,DEPS,CLF,CMF,CDF,EPSF
     3 ,XFLAP,ZFLAP,DELEPS(8,11),FPROP(2Ø,2),CYPROP(2Ø,4),PEFF(7)
     4 ,XZSS(24),XWC,ZWC,ZHEFF,RT,LT,PL,PLØ,TOP(2Ø,35),ETP(21,6Ø)

      COMMON/ARGS2/HITE,DENS


      LOGICAL ERROR
```

APPENDIX 3. SET UP PROGRAM LISTING - FTCHOO

FTCHOO.F

```
C       ! PROGRAM FOR PREPARING DATA FOR INPUT INTO CAM'S FLIGHT
C       ! SIMULATION PROGRAM.THE DATA IS READ IN FROM FTSUD FILE
C       ! AND THEN DISPLAYED ON A VDU ALONG WITH VARIABLE NAMES
C       ! CHANGES CAN BE MADE BY APPLYING THE COMMANDS INDICATED


      DIMENSION A(4Ø),B(54)
      DOUBLE PRECISION A1(4Ø),B1(54)

      DATA A1/'TTOT','TSAMP','NH','HN','DELH','NV','VN','DELV',
     1 'NPHI','PHIN','DELPHI','NPL','PLN','DELPL','NRPM',
     1 'RPMN','DELRPM','WEIGHT','NXCG','XCG %','DELXCG',
     1 'ZCG','PEFF(1)','PEFF(2)','PEFF(3)','PEFF(4)','PEFF(5)',
     1 'PEFF(6)','PEFF(7)','XT','ZT','XTH','ZTH','THSET',
     1 'MAXEP','NBLAD','PDIA','WSET','TPSET','ETAG'/

      DATA B1/'MASS','IXX','IYY','IZZ','IXZ','SW','CW'
     1 ,'BW','ST','CT','CETA','XP','ZP','CWP'
     1 ,'BFW','BTAIL','CLØ','CLAL','CDØ','CDAL','CMØ'
     1 ,'CMAL','EPSØ','EPSAL','QTOQ','AØ','A1','A2','A3','BØ','B1'
     1 ,'B2','B3','CDØT','CDLT','CMTØ','CMQW','CLP','CLXI','KWING'
     1 ,'KFUSE','NPSFI','NPSFR','MPSFI','MPSFR','CDB','GTR','BLØ',
     1 'WTR','TAPERF','XWC','ZWC','SØS','XQARTC'/

      OPEN (UNIT=1,FILE='FTSUD2.OUT',STATUS='OLD')
      OPEN (UNIT=2,FILE='FTSUD2Ø.OUT',STATUS='OLD')

C       ! PRIMARY DATA READ IN,STORED IN ARRAY A
      READ(1,*) (A(I),I=1,1Ø)
      READ(1,*) (A(I),I=11,2Ø)
      READ(1,*) (A(I),I=21,3Ø)
      READ(1,*) (A(I),I=31,4Ø)

C       ! SECONDARY DATA READ IN,STORED IN ARRAY B
      READ(1,*) (B(I),I=1,1Ø)
      READ(1,*) (B(I),I=11,2Ø)
      READ(1,*) (B(I),I=21,3Ø)
      READ(1,*) (B(I),I=31,4Ø)
      READ(1,*) (B(I),I=41,5Ø)
      READ(1,*) (B(I),I=51,54)

C       ! FLAG SETTING -  Ø.Ø FOR ARRAY A , 1.Ø FOR B
      FLAG=Ø.Ø

C       ! DISPLAY ON VDU HEADINGS,THEN NAMES AND VARIABLE VALUES
15Ø2  PRINT 1ØØ
      PRINT 1Ø1
      PRINT 1Ø2

      IF(FLAG==1.Ø) GO TO 17ØØ

      DO 1ØØØ I=1,2Ø
      J=I+2Ø
      PRINT 1Ø3,I,A1(I),A(I),J,A1(J),A(J)
1ØØØ  CONTINUE
```

```
1ØØØ  CONTINUE

15Ø4  PRINT 1Ø4
15ØØ  ACCEPT *,I,F
      IF(FLAG==1.Ø) GO TO 18ØØ
      A(I)=F
      GO TO 18Ø1
18ØØ  B(I)=F
18Ø1  CONTINUE
      IF(I .NE. O) GO TO 15ØØ
15Ø1  PRINT 1Ø5
      ACCEPT *,I
      GO TO (15Ø2,16ØØ,15Ø3) I+2


15Ø3  FLAG=1.Ø
      GO TO 15Ø2

17ØØ  DO 1ØØ1 I=1,27
      J=I+27
      PRINT 1Ø3,I,B1(I),B(I),J,B1(J),B(J)
1ØØ1  CONTINUE
      GO TO 15Ø4


c      ! PRIMARY DATA WRITTEN OUT FROM ARRAY A
16ØØ  WRITE(2,*) (A(I),I=1,1Ø)
      WRITE(2,*) (A(I),I=11,2Ø)
      WRITE(2,*) (A(I),I=21,3Ø)
      WRITE(2,*) (A(I),I=31,4Ø)

c      ! SECONDARY DATA WRITTEN OUT FROM ARRAY B
      WRITE(2,*) (B(I),I=1,1Ø)
      WRITE(2,*) (B(I),I=11,2Ø)
      WRITE(2,*) (B(I),I=21,3Ø)
      WRITE(2,*) (B(I),I=31,4Ø)
      WRITE(2,*) (B(I),I=41,5Ø)
      WRITE(2,*) (B(I),I=51,54)


1ØØ   FORMAT(/,/,9X,' **** FTSIM SETUP DATA-TO CHANGE TYPE I,F'
     1,' **** ')
1Ø1   FORMAT(/,17X,' - TERMINATE CHANGES WITH Ø,Ø - ')
1Ø2   FORMAT(/,2(' #(I)',1X,'QUANTITY',7X,'VALUE(F)',4X ))
1Ø3   FORMAT(2(1X,I2,3X,A1Ø,F14.7,2X))
1Ø4   FORMAT(/,' TO CHANGE VALUE TYPE I,NEW F ')
1Ø5   FORMAT(/,' TYPE -1,Ø,1 TO REVIEW,TERMINATE OR VIEW REMAINING'
     1,' DATA ')

      CLOSE(UNIT=1)
      CLOSE(UNIT=2)

      END
```

:FTCHOO

**** FTSIM SETUP DATA-TO CHANGE TYPE I,F ****

- TERMINATE CHANGES WITH Ø,Ø -

#(I)	QUANTITY	VALUE (F)	#(I)	QUANTITY	VALUE (F)
1	TTOT	.ØØØØØØØ	21	DELXCG	.113506Ø
2	TSAMP	.1ØØØØØØ	22	ZCG	.Ø591ØØØ
3	NH	1.ØØØØØØØ	23	PEFF (1)	1.ØØØØØØØ
4	HN	1.ØØØØØØØ	24	PEFF (2)	1.ØØØØØØØ
5	DELH	.ØØØØØØØ	25	PEFF (3)	1.ØØØØØØØ
6	NV	21.ØØØØØØØ	26	PEFF (4)	1.ØØØØØØØ
7	VN	6Ø.ØØØØØØØ	27	PEFF (5)	1.ØØØØØØØ
8	DELV	3.ØØØØØØØ	28	PEFF (6)	1.ØØØØØØØ
9	NPHI	1.ØØØØØØØ	29	PEFF (7)	.ØØØØØØØ
1Ø	PHIN	.ØØØØØØØ	3Ø	XT	3.ØØØØØØØ
11	DELPHI	1ØØ.ØØØØØØØ	31	ZT	-.55ØØØØØ
12	NPL	.ØØØØØØØ	32	XTH	-3.66ØØØØ1
13	PLN	.ØØØØØØØ	33	ZTH	.ØØØØØØØ
14	DELPL	.ØØØØØØØ	34	THSET	.ØØØØØØØ
15	NRPM	1.ØØØØØØØ	35	MAXEP	155ØØØ.ØØØØØØØ
16	RPMN	26ØØ.ØØØØØØØ	36	NBLAD	3.ØØØØØØØ
17	DELRPM	.ØØØØØØØ	37	PDIA	1.9ØØØØØØ
18	WEIGHT	125Ø.ØØØØØØØ	38	WSET	.Ø175ØØØ
19	NXCG	3.ØØØØØØØ	39	TPSET	-.ØØ36ØØØ
2Ø	XCG %	25.Ø988597	4Ø	ETAG	.ØØØØØØØ

TO CHANGE VALUE TYPE I,NEW F
4 Ø.
7 1ØØ.
Ø Ø

TYPE -1,Ø,1 TO REVIEW,TERMINATE OR VIEW REMAINING DATA
-1

**** FTSIM SETUP DATA-TO CHANGE TYPE I,F ****

- TERMINATE CHANGES WITH Ø,Ø -

#(I)	QUANTITY	VALUE (F)	#(I)	QUANTITY	VALUE (F)
1	TTOT	.ØØØØØØØ	21	DELXCG	.113506Ø
2	TSAMP	.1ØØØØØØ	22	ZCG	.Ø591ØØØ
3	NH	1.ØØØØØØØ	23	PEFF (1)	1.ØØØØØØØ
4	HN	.ØØØØØØØ	24	PEFF (2)	1.ØØØØØØØ
5	DELH	.ØØØØØØØ	25	PEFF (3)	1.ØØØØØØØ
6	NV	21.ØØØØØØØ	26	PEFF (4)	1.ØØØØØØØ
7	VN	1ØØ.ØØØØØØØ	27	PEFF (5)	1.ØØØØØØØ
8	DELV	3.ØØØØØØØ	28	PEFF (6)	1.ØØØØØØØ
9	NPHI	1.ØØØØØØØ	29	PEFF (7)	.ØØØØØØØ
1Ø	PHIN	.ØØØØØØØ	3Ø	XT	3.ØØØØØØØ
11	DELPHI	1ØØ.ØØØØØØØ	31	ZT	-.55ØØØØØ
12	NPL	.ØØØØØØØ	32	XTH	-3.66ØØØØ1
13	PLN	.ØØØØØØØ	33	ZTH	.ØØØØØØØ
14	DELPL	.ØØØØØØØ	34	THSET	.ØØØØØØØ
15	NRPM	1.ØØØØØØØ	35	MAXEP	155ØØØ.ØØØØØØØ
16	RPMN	26ØØ.ØØØØØØØ	36	NBLAD	3.ØØØØØØØ
17	DELRPM	.ØØØØØØØ	37	PDIA	1.9ØØØØØØ
18	WEIGHT	125Ø.ØØØØØØØ	38	WSET	.Ø175ØØØ
19	NXCG	3.ØØØØØØØ	39	TPSET	-.ØØ36ØØØ
2Ø	XCG %	25.Ø988597	4Ø	ETAG	.ØØØØØØØ

TO CHANGE VALUE TYPE I,NEW F
Ø Ø

TYPE -1,Ø,1 TO REVIEW,TERMINATE OR VIEW REMAINING DATA
1

**** FTSIM SETUP DATA-TO CHANGE TYPE I,F ****

- TERMINATE CHANGES WITH Ø,Ø -

#(I)	QUANTITY	VALUE(F)	#(I)	QUANTITY	VALUE(F)
1	MASS	125Ø.ØØØØØØØ	28	A2	2.2ØØØØØØ
2	IXX	1355.ØØØØØØØ	29	A3	.ØØØØØØØ
3	IYY	2466.ØØØØØØØ	3Ø	BØ	.ØØØØØØØ
4	IZZ	366Ø.ØØØØØØØ	31	B1	.ØØØØØØØ
5	IXZ	.ØØØØØØØ	32	B2	-.42ØØØØØ
6	SW	15.ØØØØØØØ	33	B3	-.12ØØØØØ
7	CW	1.57ØØØØ1	34	CDØT	.ØØØØØØØ
8	BW	1Ø.ØØØØØØØ	35	CDLT	.ØØØØØØØ
9	ST	2.72ØØØØØ	36	CMTØ	.ØØØØØØØ
1Ø	CT	.87ØØØØØ	37	CMQW	.ØØØØØØØ
11	CETA	.337ØØØØ	38	CLP	-.4Ø5ØØØØ
12	XP	-1.266ØØØØ	39	CLXI	-.36ØØØØØ
13	ZP	.26ØØØØØ	4Ø	KWING	.ØØØØØØØ
14	CWP	2.ØØØØØØØ	41	KFUSE	.ØØØØØØØ
15	BFW	.81ØØØØØ	42	NPSFI	.Ø1ØØØØØ
16	BTAIL	3.1Ø99999	43	NPSFR	.ØØØØØØØ
17	CLØ	.22561ØØ	44	MPSFI	.ØØØØØØØ
18	CLAL	4.Ø7ØØØØ2	45	MPSFR	.ØØØØØØØ
19	CDØ	.Ø38ØØØØ	46	CDB	.Ø125ØØØ
2Ø	CDAL	.Ø65ØØØØ	47	GTR	1.ØØØØØØØ
21	CMØ	-.Ø25ØØØØ	48	BLØ	25.ØØØØØØØ
22	CMAL	.ØØØØØØØ	49	WTR	.44ØØØØØ
23	EPSØ	.ØØØØØØØ	5Ø	TAPERF	.5ØØØØØØ
24	EPSAL	.4ØØØØØØ	51	XWC	-1.2ØØØØØØ
25	QTOQ	1.ØØØØØØØ	52	ZWC	.26ØØØØØ
26	AØ	.ØØØØØØØ	53	SØS	1.5ØØØØØØ
27	A1	3.16ØØØØ1	54	XQARTC	-1.2ØØØØØØ

TO CHANGE VALUE TYPE I,NEW F
Ø Ø

TYPE -1,Ø,1 TO REVIEW,TERMINATE OR VIEW REMAINING DATA
Ø
:

APPENDIX 4. EXAMPLE OF TIME HISTORIES ANALYSIS

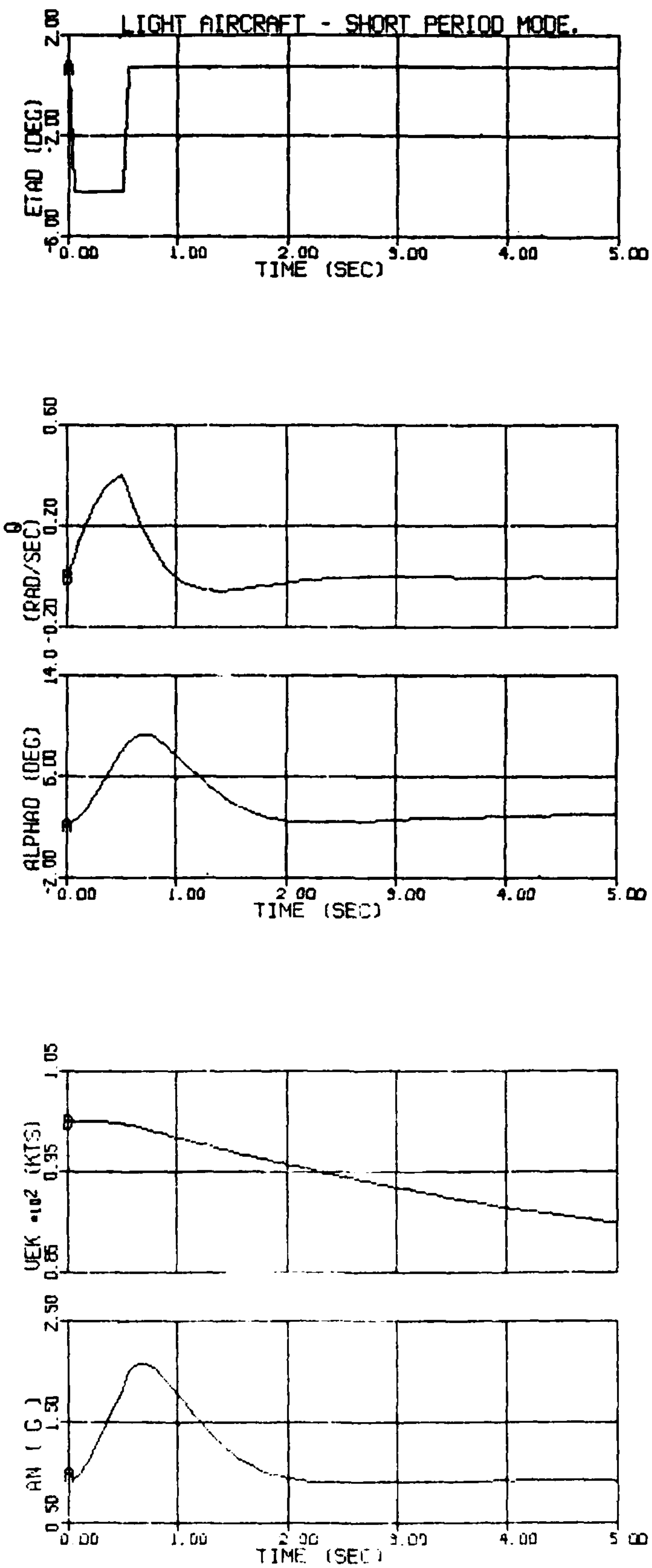
SDOFAP.SETUP

```
"*** ACSL. COMMAND FILE TO PERFORM TIME HISTORY ANALYSIS. ***"
"*** APPLICATION: LONGITUDINAL STABILITY STUDY OF A LIGHT ***"
"*** AIRCRAFT INFLUENCED BY POWER EFFECTS.                ***"

S PRN=9,TCWPRN=72,RRR=21

"*** TIME HISTORY ANALYSIS OF THE SHORT PERIOD RESPONSE ***"
"*** OF A LIGHT AIRCRAFT TO AN ELEVATOR PULSE INPUT.    ***"

S TSTP=5.Ø,NSTP=1,CINT=.Ø5
S CALPLT=.F.,PRNPLT=.F.,STRPLT=.T.,GRDSPL=.T.
OUTPUT VEK,ALPHAD,Q,ALT,ETAD,GAMMAD,THETAD,TIME,AN,'NCIOUT'=1Ø
PREPAR TIME,VEK,ALPHAD,Q,ETAD,ALT,GAMMAD,THETAD,AN

S DPL=1.
S THRMAX=Ø.Ø,TSTART=Ø.Ø,TPULSE=TSTP,TREPET=2ØØ.
S ETAMAX=-5.Ø,ESTART=Ø.Ø1,EPULSE=.5,EREPET=2ØØ.

S TSTOP=TSTP,DXCGP=7.22968
S DALTØ=1ØØØØ.

START
D CGPOS,PLS
S CMD=DIS

S RRR=21

S TITLE='   LIGHT AIRCRAFT - SHORT PERIOD MODE.          '
RANGE ETAD
S CMD=DIS
PLOT ETAD,'XHI'=TSTP,'HI'=HI1,'LO'=LO1,'XTAG'='(SEC)','TAG'='(DEG)'

S TITLE='                                                 '
RANGE ALPHAD,Q
S CMD=DIS
PLOT ALPHAD,'HI'=HI1,'LO'=LO1,'TAG'='(DEG)' ...
     ,Q,'HI'=HI2,'LO'=LO2,'TAG'='(RAD/SEC)'

RANGE AN,VEK
S CMD=DIS
PLOT AN,'HI'=HI1,'LO'=LO1,'TAG'='( G )' ...
     ,VEK,'HI'=HI2,'LO'=LO2,'TAG'='(KTS)'

S TITLE='   LIGHT AIRCRAFT - SHORT PERIOD MODE.          '
RANGE GAMMAD,THETAD
S CMD=DIS
PLOT GAMMAD,'HI'=HI1,'LO'=LO1,'TAG'='(DEG)' ...
     ,THETAD,'HI'=HI2,'LO'=LO2,'TAG'='(DEG)'

S CMD=DIS
```

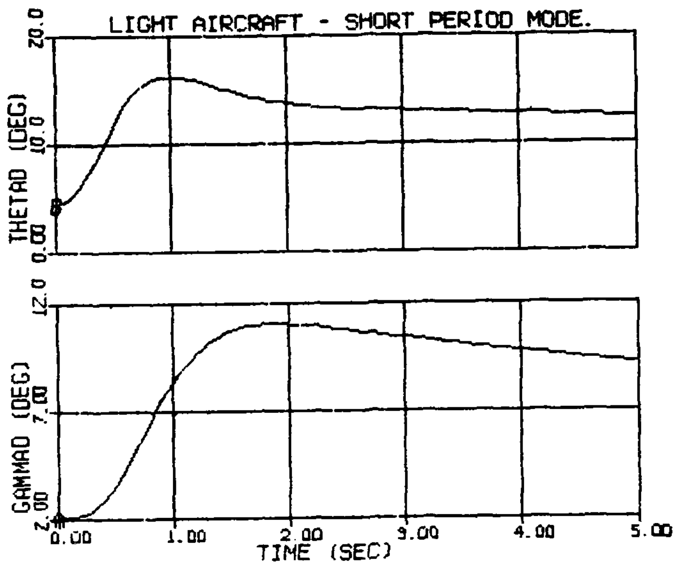
SDOFAP.L

```
--------

'*** TIME HISTORY ANALYSIS OF THE SHORT PERIOD RESPONSE ***'
'*** OF A LIGHT AIRCRAFT TO AN ELEVATOR PULSE INPUT.    ***'

S TSTP=5.Ø,NSTP=1,CINT=.Ø5
S CALPLT=.F.,PRNPLT=.F.,STRPLT=.T.,GRDSPL=.T.
OUTPUT VEK,ALPHAD,Q,ALT,ETAD,GAMMAD,THETAD,TIME,AN,'NCIOUT'=1Ø
PREPAR TIME,VEK,ALPHAD,Q,ETAD,ALT,GAMMAD,THETAD,AN

S DPL=1.
S THRMAX=Ø.Ø,TSTART=Ø.Ø,TPULSE=TSTP,TREPET=2ØØ.
S ETAMAX=-5.Ø,ESTART=Ø.Ø1,EPULSE=.5,EREPET=2ØØ.

S TSTOP=TSTP,DXCGP=7.22968
S DALTØ=1ØØØØ.

START
       VEK 99.9999Ø59      ALPHAD 2.36114Ø2Ø1            Q Ø.
       ALT 1ØØØØ.ØØØØ        ETAD Ø.8ØØ67985        GAMMAD 2.13Ø5Ø99Ø
    THETAD 4.49191191        TIME Ø.                    AN Ø.996956Ø6

       VEK 99.7383439      ALPHAD 8.Ø85758ØØ             Q Ø.4Ø193779
       ALT 1ØØØ4.2341        ETAD-4.19923639        GAMMAD 3.59332151
    THETAD 11.679Ø795        TIME Ø.5ØØØØØØØ            AN 1.847533Ø8

       VEK 98.415Ø283      ALPHAD 7.8256Ø635             Q-Ø.ØØ434133
       ALT 1ØØ14.512Ø        ETAD Ø.8ØØ67985        GAMMAD 8.39623514
    THETAD 16.2218415        TIME 1.ØØØØØØØØ            AN 1.7918Ø178

       VEK 97.ØØ847Ø9      ALPHAD 4.13979811             Q-Ø.Ø563447Ø
       ALT 1ØØ3Ø.8854        ETAD Ø.8ØØ67985        GAMMAD 1Ø.7648666
    THETAD 14.9Ø46647        TIME 1.5ØØØØØØØ            AN 1.19691563

       VEK 95.7332492      ALPHAD 2.6Ø16322Ø             Q-Ø.Ø2671Ø67
       ALT 1ØØ49.Ø153        ETAD Ø.8ØØ67985        GAMMAD 11.1ØØ8773
    THETAD 13.7Ø25Ø95        TIME 2.ØØØØØØØØ            AN Ø.94981Ø21

       VEK 94.55Ø3649      ALPHAD 2.43318653             Q-Ø.ØØ849723
       ALT 1ØØ66.8256        ETAD Ø.8ØØ67985        GAMMAD 1Ø.8Ø832ØØ
    THETAD 13.2415Ø65        TIME 2.5ØØØØØØØ            AN Ø.9Ø641519

       VEK 93.4366668      ALPHAD 2.62361519             Q-Ø.ØØ4415Ø9
       ALT 1ØØ83.8828        ETAD Ø.8ØØ67985        GAMMAD 1Ø.4554629
    THETAD 13.Ø79Ø781        TIME 3.ØØØØØØØØ            AN Ø.91313888

       VEK 92.381Ø44Ø      ALPHAD 2.8Ø481571             Q-Ø.ØØ586753
       ALT 1Ø1ØØ.2189        ETAD Ø.8ØØ67985        GAMMAD 1Ø.1325532
    THETAD 12.9373689        TIME 3.5ØØØØØØØ            AN Ø.918Ø9176

       VEK 91.43Ø6631      ALPHAD 2.92349396             Q-Ø.ØØ661579
       ALT 1Ø115.8837        ETAD Ø.8ØØ67985        GAMMAD 9.819ØØ547
    THETAD 12.7424994        TIME 4.ØØØØØØØØ            AN Ø.92266517

       VEK 9Ø.655Ø76Ø      ALPHAD 3.Ø4215Ø94             Q-Ø.ØØ63Ø872
       ALT 1Ø13Ø.9422        ETAD Ø.8ØØ67985        GAMMAD 9.52346617
    THETAD 12.5656171        TIME 4.5ØØØØØØØ            AN Ø.923712Ø1

       VEK 89.93Ø8754      ALPHAD 3.13963864             Q-Ø.ØØ79ØØ29
       ALT 1Ø145.4264        ETAD Ø.8ØØ67985        GAMMAD 9.224Ø6554
    THETAD 12.3637Ø42        TIME 5.ØØØØØØØØ            AN Ø.92242613
```

```
 D CGPOS,PLS
      CGPOS 32.3285397         PLS 1.ØØØØØØØ
 S CMD=DIS
S CMD=1Ø

 S RRR=21

 S TITLE='   LIGHT AIRCRAFT - SHORT PERIOD MODE.          '
 RANGE ETAD
       ETAD-4.19923639  Ø.8ØØ67985
 S CMD=DIS
S HI1=2.,LO1=-6.
S CMD=1Ø
 PLOT ETAD,'XHI'=TSTP,'HI'=HI1,'LO'=LO1,'XTAG'='(SEC)','TAG'='(DEG)'

 S TITLE='                                                  '
 RANGE ALPHAD,Q
     ALPHAD 2.3614Ø2Ø1  9.38391Ø14
          Q-Ø.Ø5862956  Ø.4Ø193779
 S CMD=DIS
S HI1=14.,LO1=-2.,HI2=.6,LO2=-.2
S CMD=1Ø
 PLOT ALPHAD,'HI'=HI1,'LO'=LO1,'TAG'='(DEG)' ...
     ,Q,'HI'=HI2,'LO'=LO2,'TAG'='(RAD/SEC)'

 RANGE AN,VEK
         AN Ø.9Ø6315Ø5  2.Ø8298743
        VEK 89.8613617  1ØØ.ØØØ728
 S CMD=DIS
S HI1=2.5,LO1=.5,HI2=1Ø5.,LO2=85.
S CMD=1Ø
 PLOT AN,'HI'=HI1,'LO'=LO1,'TAG'='( G )' ...
     ,VEK,'HI'=HI2,'LO'=LO2,'TAG'='(KTS)'

 S TITLE='   LIGHT AIRCRAFT - SHORT PERIOD MODE.          '
 RANGE GAMMAD,THETAD
     GAMMAD 2.Ø9Ø2Ø455  11.1164848
     THETAD 4.49191191  16.2218415
 S CMD=DIS
S HI1=12.,LO1=2.,HI2=2Ø.,LO2=Ø.
S CMD=1Ø
 PLOT GAMMAD,'HI'=HI1,'LO'=LO1,'TAG'='(DEG)' ...
     ,THETAD,'HI'=HI2,'LO'=LO2,'TAG'='(DEG)'

 S CMD=DIS
STOP
```



LIGHT AIRCRAFT - SHORT PERIOD MODE.
ETAD (DEG)
2.00
-2.00
-6.00
0.00
1.00
2.00
3.00
4.00
5.00
TIME (SEC)
Q (RAD/SEC)
0.60
0.20
-0.20
ALPHAD (DEG)
14.0
6.00
-2.00
0.00
1.00
2.00
3.00
4.00
5.00
TIME (SEC)
VEK *10² (KTS)
1.05
0.95
0.85
AN (G)
2.50
1.50
0.50
0.00
1.00
2.00
3.00
4.00
5.00
TIME (SEC)


LIGHT AIRCRAFT - SHORT PERIOD MODE.
THETAD (DEG)
20.0
10.0
0.00
GAMMAD (DEG)
12.0
7.00
2.00
0.00
1.00
2.00
3.00
4.00
5.00
TIME (SEC)

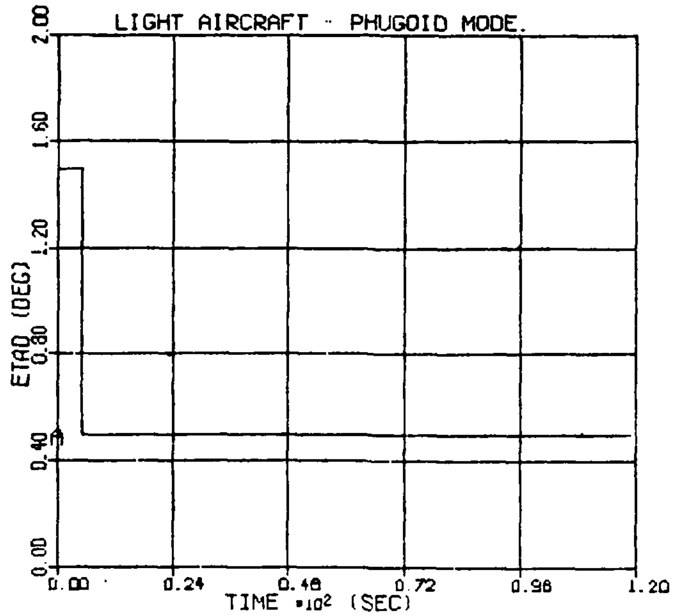
SDOFAP.SETUP

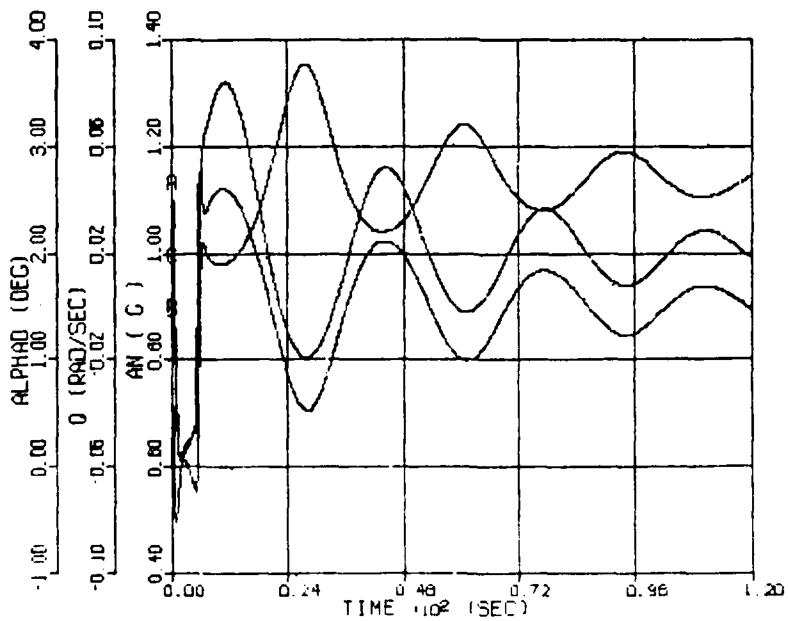
```
"*** ACSL. COMMAND FILE TO PERFORM TIME HISTORY ANALYSIS. ***"
"*** APPLICATION: LONGITUDINAL STABILITY STUDY OF A LIGHT ***"
"*** AIRCRAFT INFLUENCED BY POWER EFFECTS.               ***"

S PRN=9,TCWPRN=72,RRR=21

"*** TIME HISTORY ANALYSIS OF THE PHUGOID RESPONSE OF ***"
"*** A LIGHT AIRCRAFT TO AN ELEVATOR PULSE INPUT.     ***"

S TSTP=12Ø.Ø,NSTP=1,CINT=.Ø5
S CALPLT=.T.,PRNPLT=.F.,STRPLT=.F.,GRDCPL=.T.
OUTPUT VEK,ALPHAD,Q,ALT,ETAD,GAMMAD,THETAD,TIME,AN,'NCIOUT'=24Ø
PREPAR TIME,VEK,ALPHAD,Q,ETAD,ALT,GAMMAD,THETAD,AN

S DPL=Ø.
S THRMAX=Ø.Ø,TSTART=Ø.Ø,TPULSE=TSTP,TREPET=2ØØ.
S ETAMAX=1.Ø,ESTART=Ø.Ø1,EPULSE=5.,EREPET=2ØØ.

S TSTOP=TSTP,DXCGP=7.22968
S DALTØ=1ØØØØ.

START
D CGPOS,PLS
S CMD=DIS

S RRR=21

S TITLE='   LIGHT AIRCRAFT - PHUGOID MODE.                  '
RANGE ETAD
S CMD=DIS
PLOT ETAD,'XHI'=TSTP,'HI'=HI1,'LO'=LO1,'XTAG'='(SEC)','TAG'='(DEG)'

S TITLE='                                                   '
RANGE ALPHAD,Q,AN
S CMD=DIS
PLOT ALPHAD,'HI'=HI1,'LO'=LO1,'TAG'='(DEG)',Q,'HI'=HI2,'LO'=LO2 ...
           ,'TAG'='(RAD/SEC)',AN,'HI'=HI3,'LO'=LO3,'TAG'='( G )'

RANGE VEK,GAMMAD,THETAD
S CMD=DIS
PLOT VEK,'HI'=HI1,'LO'=LO1,'TAG'='(KTS)',GAMMAD,'HI'=HI2,'LO'=LO2 ...
        ,'TAG'='(DEG)',THETAD,'HI'=HI3,'LO'=LO3,'TAG'='(DEG)'

S CMD=DIS
```

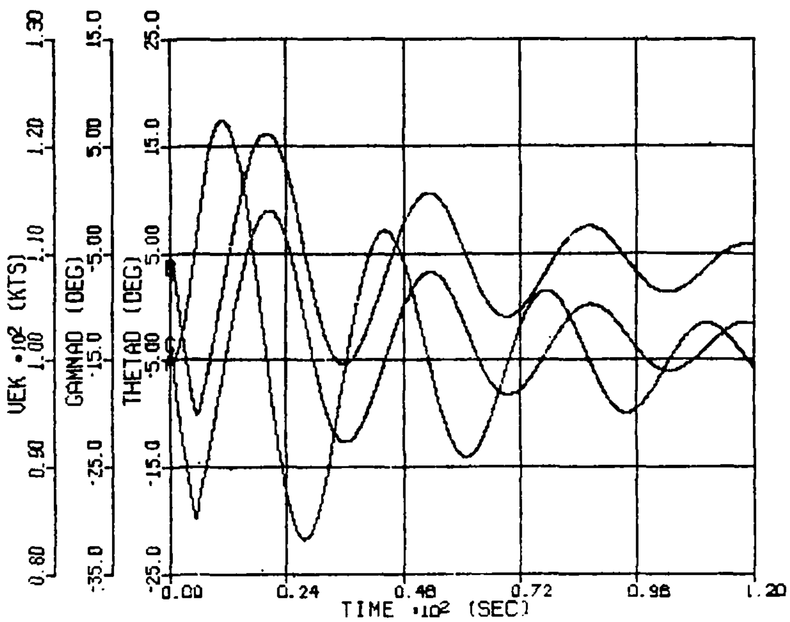
SDOFAP.L

```
'*** TIME HISTORY ANALYSIS OF THE PHUGOID RESPONSE OF ***'
'*** A LIGHT AIRCRAFT TO AN ELEVATOR PULSE INPUT.     ***'

S TSTP=12Ø.Ø,NSTP=1,CINT=.Ø5
S CALPLT=.T.,PRNPLT=.F.,STRPLT=.F.,GRDCPL=.T.
OUTPUT VEK,ALPHAD,Q,ALT,ETAD,GAMMAD,THETAD,TIME,AN,'NCIOUT'=24Ø
PREPAR TIME,VEK,ALPHAD,Q,ETAD,ALT,GAMMAD,THETAD,AN

S DPL=Ø.
S THRMAX=Ø.Ø,TSTART=Ø.Ø,TPULSE=TSTP,TREPET=2ØØ.
S ETAMAX=1.Ø,ESTART=Ø.Ø1,EPULSE=5.,EREPET=2ØØ.

S TSTOP=TSTP,DXCGP=7.22968
S DALTØ=1ØØØØ.

START
       VEK 99.9999Ø59      ALPHAD 2.69495477           Q Ø.
       ALT 1ØØØØ.ØØØØ        ETAD Ø.49637286      GAMMAD-6.12632886
    THETAD-3.431374Ø9        TIME Ø.                  AN Ø.99929857

       VEK 121.937459      ALPHAD 1.93136242           Q Ø.Ø4324758
       ALT 9399.4Ø124        ETAD Ø.49637286      GAMMAD-5.39135482
    THETAD-3.45999241        TIME 12.ØØØØØØØ          AN 1.315Ø81Ø5

       VEK 87.552Ø883      ALPHAD 3.43412635           Q-Ø.Ø254985Ø
       ALT 9521.73771        ETAD Ø.49637286      GAMMAD 2.8Ø32277Ø
    THETAD 6.237354Ø5        TIME 24.ØØØØØØØ          AN Ø.85162Ø12

       VEK 99.1196767      ALPHAD 2.65628993           Q-Ø.ØØ1Ø2716
       ALT 9218.32978        ETAD Ø.49637286      GAMMAD-15.26ØØ428
    THETAD-12.6Ø37528        TIME 36.ØØØØØØØ          AN Ø.97578389

       VEK 1Ø9.29Ø973      ALPHAD 2.31813959           Q Ø.Ø1964675
       ALT 8795.32924        ETAD Ø.49637286      GAMMAD-2.298Ø4627
    THETAD Ø.Ø2ØØ9332        TIME 48.ØØØØØØØ          AN 1.126Ø1581

       VEK 91.1228162      ALPHAD 3.198Ø2476           Q-Ø.Ø1958144
       ALT 8763.29354        ETAD Ø.49637286      GAMMAD-4.24752148
    THETAD-1.Ø4949672        TIME 6Ø.ØØØØØØØ          AN Ø.8927Ø2Ø9

       VEK 1Ø3.2Ø91Ø2      ALPHAD 2.51945226           Q Ø.ØØ7Ø2534
       ALT 8414.Ø1146        ETAD Ø.49637286      GAMMAD-1Ø.252ØØ7Ø
    THETAD-7.73255472        TIME 72.ØØØØØØØ          AN 1.Ø3622142

       VEK 1Ø2.264847      ALPHAD 2.59925787           Q Ø.ØØ533936
       ALT 815Ø.97446        ETAD Ø.49637286      GAMMAD-2.7237836Ø
    THETAD-Ø.12452573        TIME 84.ØØØØØØØ          AN 1.Ø3Ø132Ø3

       VEK 95.5421614      ALPHAD 2.92362864           Q-Ø.ØØ98Ø669
       ALT 8ØØØ.27666        ETAD Ø.49637286      GAMMAD-6.8296476Ø
    THETAD-3.9Ø6Ø1896        TIME 96.ØØØØØØØ          AN Ø.94358945

       VEK 1Ø3.12Ø145      ALPHAD 2.54128177           Q Ø.ØØ67486Ø
       ALT 7681.26595        ETAD Ø.49637286      GAMMAD-7.28543841
    THETAD-4.74415664        TIME 1Ø8.ØØØØØØ          AN 1.Ø38Ø18Ø2

       VEK 99.3821384      ALPHAD 2.73222594           Q-Ø.ØØ1Ø1423
       ALT 747Ø.985Ø9        ETAD Ø.49637286      GAMMAD-4.17Ø6Ø998
    THETAD-1.438384Ø4        TIME 12Ø.ØØØØØØ          AN Ø.9926Ø128
```

```
 D CGPOS,PLS
      CGPOS 32.3285397         PLS Ø.
 S CMD=DIS
S CMD=1Ø

 S RRR=21

 S TITLE='   LIGHT AIRCRAFT - PHUGOID MODE.                    '
 RANGE ETAD
       ETAD Ø.49637286  1.49635611
 S CMD=DIS
S HI1=2.,LO1=Ø.
S CMD=1Ø
 PLOT ETAD,'XHI'=TSTP,'HI'=HI1,'LO'=LO1,'XTAG'='(SEC)','TAG'='(DEG)'

 S TITLE='                                                     '
 RANGE ALPHAD,Q,AN
     ALPHAD-Ø.23557132  3.77131611
          Q-Ø.Ø8Ø342Ø3  Ø.Ø5ØØ5342
         AN Ø.617Ø2932  1.32Ø9Ø147
 S CMD=DIS
S HI1=4.,LO1=-1.,HI2=.1,LO2=-.1,HI3=1.4,LO3=.4
S CMD=1Ø
 PLOT ALPHAD,'HI'=HI1,'LO'=LO1,'TAG'='(DEG)',Q,'HI'=HI2,'LO'=LO2 ...
           ,'TAG'='(RAD/SEC)',AN,'HI'=HI3,'LO'=LO3,'TAG'='( G )'

 RANGE VEK,GAMMAD,THETAD
        VEK 83.2781743  122.456445
     GAMMAD-2Ø.Ø4Ø4889  6.219160ØЗ3
     THETAD-19.6957197  8.959ØØØ22
 S CMD=DIS
S HI1=13Ø.,LO1=8Ø.,HI2=15.,LO2=-35.,HI3=25.,LO3=-25.
S CMD=1Ø
 PLOT VEK,'HI'=HI1,'LO'=LO1,'TAG'='(KTS)',GAMMAD,'HI'=HI2,'LO'=LO2 ...
        ,'TAG'='(DEG)',THETAD,'HI'=HI3,'LO'=LO3,'TAG'='(DEG)'

 S CMD=DIS
STOP
```


LIGHT AIRCRAFT - PHUGOID MODE.
ETAD (DEG)
2.00
1.60
1.20
0.80
0.40
0.00
0.00
0.24
0.48
0.72
0.96
1.20
TIME *10² (SEC)


ALPHAD (DEG)
Q (RAD/SEC)
AN (G)
4.00
3.00
2.00
1.00
0.00
-1.00
0.10
0.06
0.02
-0.02
-0.06
-0.10
1.40
1.20
1.00
0.80
0.60
0.40
0.00
0.24
0.48
0.72
0.96
1.20
TIME *10² (SEC)



VEK •10² (KTS)
GAMMAD (DEG)
THETAD (DEG)
TIME •10² (SEC)
0.00
0.24
0.48
0.72
0.96
1.20
0.80
0.90
1.00
1.10
1.20
1.30
-35.0
-25.0
-15.0
-5.00
5.00
15.0
-25.0
-15.0
-5.00
5.00
15.0
25.0

APPENDIX 5. EXAMPLE OF EIGEN ANALYSIS

SDOFAP.SETUP

```
"*** ACSL. COMMAND FILE TO PERFORM EIGEN ANALYSIS. ***"
"*** APPLICATION: LONGITUDINAL STABILITY STUDY OF  ***"
"*** A LIGHT AIRCRAFT INFLUENCED BY POWER EFFECTS. ***"


S PRN=9,TCWPRN=72

"*** EIGEN ANALYSIS OF LONGITUDINAL MOTION OF A  ***"
"*** LIGHT AIRCRAFT INFLUENCED BY POWER EFFECTS. ***"

S NSTP=1,CINT=.Ø5
S DXCGP=7.22968,TSTOP=Ø.
S DALTØ=1ØØØØ.

PREPAR TIME,VEK,ALPHAD,Q,ETAD,ALT,GAMMAD,THETAD,AN


"*** EIGEN ANALYSIS ***"

S DPL=Ø.

START
D CGPOS,PLS

ANALYZ 'FREEZE'=X,Y,Z,P,R,BETAR,TAU1,TAU3
ANALYZ 'EIGVEC'=.T.,'EIGEN'


"*** EIGEN ANALYSIS ***"

S BEGIN=.F.
S DPL=1.

START
D CGPOS,PLS

ANALYZ 'EIGVEC'=.T.,'EIGEN'

STOP
```

SDOFAP.L

```
'*** EIGEN ANALYSIS OF LONGITUDINAL MOTION OF A  ***'
'*** LIGHT AIRCRAFT INFLUENCED BY POWER EFFECTS. ***'

S NSTP=1,CINT=.Ø5
S DXCGP=7.22968,TSTOP=Ø.
S DALTØ=1ØØØØ.

PREPAR TIME,VEK,ALPHAD,Q,ETAD,ALT,GAMMAD,THETAD,AN


'*** EIGEN ANALYSIS ***'

S DPL=Ø.

START
D CGPOS,PLS
     CGPOS 32.3285397          PLS Ø.

ANALYZ 'FREEZE'=X,Y,Z,P,R,BETAR,TAU1,TAU3
ANALYZ 'EIGVEC'=.T.,'EIGEN'


COMPLEX EIGEN VALUES IN ASCENDING ORDER
        REAL           IMAGINARY       FREQUENCY     DAMPING
  1 -3.2785E-16
  2 -Ø.Ø1886ØØ2  +/-Ø.18552241      Ø.186479     Ø.1Ø1138
  4 -1.84179949  +/-1.7122937Ø      2.514791     Ø.732387

COMPLEX EIGEN VECTORS
                  1                       2                        3
  1 -Ø.9819847 Ø.          2.849E-Ø4-3.8Ø6E-Ø5     2.849E-Ø4 3.8Ø6E-Ø5
  2  Ø.Ø29Ø791 Ø.          Ø.ØØ95129-Ø.ØØ127Ø7     Ø.ØØ95129 Ø.ØØ127Ø7
  3  1.461E-16 Ø.          1.127E-Ø4 Ø.ØØ35793     1.127E-Ø4-Ø.ØØ35793
  4  Ø.1867Ø97 Ø.          Ø.Ø142453 Ø.9998451     Ø.Ø142453-Ø.9998451
  5  2.8ØØE-Ø7 Ø.          7.715E-Ø5-Ø.ØØ137Ø8     7.715E-Ø5 Ø.ØØ137Ø8

                  4                       5
     Ø.ØØ25Ø64 Ø.ØØ31429   Ø.ØØ25Ø64-Ø.ØØ31429
     Ø.Ø836763 Ø.1Ø49257   Ø.Ø836763-Ø.1Ø49257
     Ø.Ø5112ØØ-Ø.6733628   Ø.Ø5112ØØ Ø.6733628
     Ø.4396764 Ø.436932Ø   Ø.4396764-Ø.436932Ø
     Ø.3589739 Ø.1133794   Ø.3589739-Ø.1133794


'*** EIGEN ANALYSIS ***'

S BEGIN=.F.
S DPL=1.

START
D CGPOS,PLS
     CGPOS 32.3285397          PLS 1.ØØØØØØØ

ANALYZ 'EIGVEC'=.T.,'EIGEN'
```

```
COMPLEX EIGEN VALUES IN ASCENDING ORDER
          REAL          IMAGINARY        FREQUENCY     DAMPING
   1 -6.9337E-17
   2 -0.01700625  +/-0.15272301      0.153667     0.110670
   4 -1.99230691  +/-1.74483906      2.648349     0.752283


COMPLEX EIGEN VECTORS
                  1                      2                       3
   1 -0.9992318 0.           -1.808E-04 2.549E-04   -1.808E-04-2.549E-04
   2 -0.0391893 0.            0.0046099-0.0064994    0.0046099 0.0064994
   3 -1.728E-16 0.            0.0018298 0.0016304    0.0018298-0.0016304
   4 -4.805E-09 0.            0.6663900 0.7455544    0.6663900-0.7455544
   5  4.605E-12 0.           -0.0012826-0.0013281   -0.0012826 0.0013281


                  4                      5
      3.778E-04-0.0052915     3.778E-04 0.0052915
     -0.0096333 0.1349206    -0.0096333-0.1349206
      0.5096059-0.5043772     0.5096059 0.5043772
     -0.0298649 0.5624622    -0.0298649-0.5624622
      0.1840649 0.3412410     0.1840649-0.3412410

  STOP

     2 WORDS TABLE SPACE USED
```

APPENDIX 6. EXAMPLE OF JACOBIAN ANALYSIS

SDOFAP.SETUP

```
"*** ACSL. COMMAND FILE TO PERFORM JACOBIAN ANALYSIS. ***"
"*** APPLICATION:  LONGITUDINAL STABILITY STUDY OF A  ***"
"*** LIGHT AIRCRAFT INFLUENCED BY POWER EFFECTS.      ***"


S PRN=9,TCWPRN=72

"*** JACOBIAN ANALYSIS TO DEDUCE NON-DIMENSIONAL ***"
"*** AERODYNAMIC DERIVATIVES OF A LIGHT AIRCRAFT ***"
"*** INFLUENCED BY POWER EFFECTS.                ***"

S NSTP=1,CINT=.Ø5
S DXCGP=7.22968,TSTOP=Ø.
S DALTØ=1ØØØØ.

PREPAR TIME,VEK,ALPHAD,Q,ETAD,ALT,GAMMAD,THETAD,AN



"*** JACOBIAN ANALYSIS ***"

S DPL=1.

START
D CGPOS,PLS

ANALYZ 'FREEZE'=X,Y,Z,P,R,BETAR,TAU1,TAU3
ANALYZ 'JACOB'

S PRN=DIS
ANALYZ 'EIGEN' $"***  THIS COMMAND IS USED ONLY TO  ***"
               $"*** INITIATE THE JACOBIAN ANALYSIS ***"

S PRN=9

"*** JACOBIAN ANALYSIS ***"

S BEGIN=.F.
S DPL=1.

START
D CGPOS,PLS

ANALYZ 'JACOB'

S PRN=DIS
ANALYZ 'EIGEN'

STOP
```

SDOFAP.L

```
'*** JACOBIAN ANALYSIS TO DEDUCE NON-DIMENSIONAL ***'
'*** AERODYNAMIC DERIVATIVES OF A LIGHT AIRCRAFT ***'
'*** INFLUENCED BY POWER EFFECTS.                ***'

S NSTP=1,CINT=.Ø5
S DXCGP=7.22968,TSTOP=Ø.
S DALTØ=1ØØØØ.

PREPAR TIME,VEK,ALPHAD,Q,ETAD,ALT,GAMMAD,THETAD,AN


'*** JACOBIAN ANALYSIS ***'

S DPL=1.

START
D CGPOS,PLS
     CGPOS 32.3285397          PLS 1.ØØØØØØØØ

ANALYZ 'FREEZE'=X,Y,Z,P,R,BETAR,TAU1,TAU3
ANALYZ 'JACOB'


ROW VECTOR NAMES
       TAUØ       1            TAU2       2                Q       3
         VT       4          ALPHAR       5

COLUMN VECTOR NAMES
     TAUØDT       1          TAU2DT       2             QDOT       3
        DVT       4          DALPHR       5

MATRIX ELEMENTS - ROWS ACROSS, COLUMNS DOWN
          1            2            3            4            5
   1  Ø.           Ø.          -Ø.Ø195946   Ø.           Ø.
   2  Ø.           Ø.           Ø.4996159   Ø.           Ø.
   3 -3.991E-Ø4    Ø.Ø1Ø1762   -2.5Ø35555  -9.349E-Ø5   -3.36526Ø7
   4  Ø.7681246  -19.585314   -Ø.Ø17429Ø  -Ø.Ø43141Ø    4.6697616
   5  4.738E-Ø4   -Ø.Ø12Ø811    Ø.9798915  -Ø.ØØ51Ø29  -1.4719298


S PRN=DIS


*** DIMENSIONAL JACOBIAN. ***


   -.Ø43141Ø     4.6697616    -.Ø17429Ø  -19.5853136

   -.ØØ51Ø29    -1.4719298     .9798915    -.Ø12Ø811

   -.ØØØØ935    -3.36526Ø7   -2.5Ø35555     .Ø1Ø1762

    .ØØØØØØØ     .ØØØØØØØ      .4996159     .ØØØØØØØ
```

NOTE: ASSUMED RELATIONSHIPS IN USE FOR CALCULATION
OF ANGULAR RATE DERIVATIVES.

*** NON-DIMENSIONAL JACOBIAN. ***

-.ØØØ5653	.ØØ1Ø215	-.Ø17429Ø	-.ØØ21422
-.ØØ4ØØ57	-.Ø192886	.9798915	-.ØØØØ792
-.ØØØØØ1Ø	-.ØØØ5779	-.Ø328Ø73	.ØØØØØØ9
.ØØØØØØØ	.ØØØØØØØ	.4996159	.ØØØØØØØ

*** NON-DIMENSIONAL AERODYNAMIC DERIVATIVES. ***

```
CTV      =    -.22Ø8641
CLV      =    -.Ø6Ø6333
CDV      =    -.Ø5Ø9233
CMV      =    -.Ø15Ø985
CLALPHA  =    4.4965Ø35
CDALPHA  =     .263286Ø
CMALPHA  =    -.2876742
CLDALPHA =    1.3753121
CMDALPHA =   -3.6791788
CLQ      =    3.2512172
CMQ      =   -8.6975235
```

```
'*** JACOBIAN ANALYSIS ***'

S BEGIN=.F.
S DPL=1.

START
D CGPOS,PLS
     CGPOS 32.3285397         PLS 1.ØØØØØØØ

ANALYZ 'JACOB'
```

```
ROW VECTOR NAMES
      TAUØ       1           TAU2        2              Q       3
        VT       4         ALPHAR        5

COLUMN VECTOR NAMES
    TAUØDT       1         TAU2DT        2           QDOT       3
       DVT       4         DALPHR        5

MATRIX ELEMENTS - ROWS ACROSS, COLUMNS DOWN
        1            2            3            4            5
   1 Ø.           Ø.          -Ø.Ø195946   Ø.           Ø.
   2 Ø.           Ø.           Ø.4996159   Ø.           Ø.
   3 -3.991E-Ø4   Ø.Ø1Ø1762   -2.5Ø35555  -9.349E-Ø5   -3.36526Ø7
   4 Ø.7681246  -19.585314    -Ø.Ø17429Ø  -Ø.Ø43141Ø    4.6697616
   5 4.738E-Ø4   -Ø.Ø12Ø811    Ø.9798915  -Ø.ØØ51Ø29   -1.4719298

 S PRN=DIS
```

*** DIMENSIONAL JACOBIAN. ***

-.Ø43141Ø	4.6697616	-.Ø17429Ø	-19.5853136
-.ØØ51Ø29	-1.4719298	.9798915	-.Ø12Ø811
-.ØØØØ935	-3.36526Ø7	-2.5Ø35555	.Ø1Ø1762
.ØØØØØØØ	.ØØØØØØØ	.4996159	.ØØØØØØØ

NOTE: ANGULAR RATE DERIVATIVES CALCULATED FROM
ELEMENTS OF THE JACOBIAN MATRIX.

BEWARE OF POSSIBLE INACCURACY ASSOCIATED WITH SMALL CLIMB ANGLES.

*** NON-DIMENSIONAL JACOBIAN. ***

-.ØØØ5653	.ØØ1Ø215	-.Ø17429Ø	-.ØØ21422
-.ØØ4ØØ57	-.Ø192886	.9798915	-.ØØØØ792
-.ØØØØØ1Ø	-.ØØØ5779	-.Ø328Ø73	.ØØØØØØ9
.ØØØØØØØ	.ØØØØØØØ	.4996159	.ØØØØØØØ

*** NON-DIMENSIONAL AERODYNAMIC DERIVATIVES. ***

CTV	=	-.22Ø8641
CLV	=	-.Ø616Ø25
CDV	=	-.Ø5Ø9233
CMV	=	-.Ø169414
CLALPHA	=	4.4918363
CDALPHA	=	.263286Ø
CMALPHA	=	-.2965481
CLDALPHA	=	1.1333446
CMDALPHA	=	-4.1392378
CLQ	=	3.6Ø67239
CMQ	=	-8.2467156

APPENDIX 7. JACOBIAN ANALYSIS SUBROUTINES

```
      SUBROUTINE INTERM(A)

C     "***** PROVIDES JACOBIAN REDUCTION ROUTINE 'ZZREDC' WITH *****"
C     "*****              FLIGHT CONDITION DATA.               *****"
C     "*****  THIS SUBROUTINE MUST BE APPENDED TO SDOFAP.ACSL  *****"

      INTEGER   ROCOL(5)
      DIMENSION A(5,5),AA(5,5),AAA(5,5)
      DIMENSION B(15)
      LOGICAL   LINEAR
      CHARACTER*1 ANS
$
      DATA ROCOL/4,5,3,2,1/
      LINEAR = .FALSE.

C     "*** REARRANGE ROWS OF JACOBIAN. ***"

      DO 1Ø I=1,5
      DO 2Ø J=1,5
      AAA(I,J)=A(ROCOL(I),J)
2Ø    CONTINUE
1Ø    CONTINUE

C     "*** REARRANGE COLS OF JACOBIAN. ***"

      DO 3Ø I=1,5
      DO 4Ø J=1,5
      AA(I,J)=AAA(I,ROCOL(J))
4Ø    CONTINUE
3Ø    CONTINUE

C     "*** WRITE DIMENSIONAL JACOBIAN TO FILE. ***"

      WRITE (9,8Ø)

      DO 7Ø I=1,4

      WRITE (9,85) (AA(I,J),J=1,4)

7Ø    CONTINUE

C     "*** ASCERTAIN WHETHER LINEAR ANALYSIS IS TO BE USED ***"

      WRITE (6,13Ø)
      READ (5,1ØØ) ANS
      IF ( ANS .EQ. 1HY .OR. ANS .EQ. 1Hy , LINEAR = .TRUE.

      IF ( LINEAR ) THEN
      WRITE(9,123)
      ELSE
      WRITE(9,124)
      WRITE(9,125)
      WRITE(9,126)
      ENDIF
```

```
C     "*** CALCULATE NON-DIMENSIONALISING FACTORS ***"

C     "*** REQUIRED A/C DATA AT EQUILIBRIUM POINT ***"

      CALL ACINFO(CTPRP,SW,CW,THSET,XLT,XEPSAL)

         QDS=.5*RHO*VT**2*SW
         CTE=CTPRP
         CWE=MASS*G/QDS
         CDE=CTE*COS(THSET)-CWE*SIN(GAMMAR)
         CLE=CWE*COS(GAMMAR)-CTE*SIN(THSET)
         AMU=MASS/(.5*RHO*SW*CW)
         FACT1=.5*CW/VT
         FACT2=.5*CW/VT/VT
         FACT3=FACT1**2
         FACT4=CW*CW/4./VT
         FACT5=CW/2.
         FACT6=RHO*SW*CW**3/8./IYY
         IYYND=1/FACT6

C     "*** NON-DIMENSIONALISE JACOBIAN ELEMENTS. ***"

         AA(1,1)=AA(1,1)*FACT1
         AA(1,2)=AA(1,2)*FACT2
         AA(1,3)=AA(1,3)
         AA(1,4)=AA(1,4)*FACT2*.5

         AA(2,1)=AA(2,1)*FACT5
         AA(2,2)=AA(2,2)*FACT1
         AA(2,3)=AA(2,3)
         AA(2,4)=AA(2,4)*FACT1*.5

         AA(3,1)=AA(3,1)*FACT4
         AA(3,2)=AA(3,2)*FACT3
         AA(3,3)=AA(3,3)*FACT1
         AA(3,4)=AA(3,4)*FACT3*.5

C     "*** ELEMENTS OF ROW 4 ARE OF NO FURTHER USE ***"

C     "*** PASS A/C INFO TO SUBROUTINE IN ARRAY B ***"

         B(1)=SIN(GAMMAR)
         B(2)=COS(GAMMAR)
         B(3)=SIN(THSET)
         B(4)=COS(THSET)
         B(5)=CWE
         B(6)=CLE
         B(7)=CTE
         B(8)=CDE
         B(9)=AMU
         B(1Ø)=IYYND
         B(11)=XLT
         B(12)=XEPSAL
         B(13)=CW

C     "****** CALCULATE NON-DIMENSIONAL DERIVATIVES ******"

      CALL ZZREDC(AA,B,LINEAR)
```

```
80    FORMAT(////,31H  *** DIMENSIONAL JACOBIAN. ***,//)
85    FORMAT(/,4X,F11.7,2X,F11.7,2X,F11.7,2X,F11.7)
100   FORMAT (A1)
123   FORMAT(//,1X,51HNOTE: ASSUMED RELATIONSHIPS IN USE FOR CALCULATION
     .       ,/,1X,35H      OF ANGULAR RATE DERIVATIVES. ,//)
124   FORMAT(//,1X,47HNOTE: ANGULAR RATE DERIVATIVES CALCULATED FROM
     .       ,/,1X,38H      ELEMENTS OF THE JACOBIAN MATRIX.)
125   FORMAT(//,1X,45HBEWARE OF POSSIBLE INACCURACY ASSOCIATED WITH
     .          ,20H SMALL CLIMB ANGLES. )
126   FORMAT(   1X,45H---------------------------------------------
     .          ,20H-------------------- ,//)
130   FORMAT(//,1X,46H SHOULD THE ANGULAR RATE DERIVATIVES BE FOUND ,/,
     .          1X,42H BY USING ASSUMED RELATIONSHIPS. (Y/N) : ?,4X$)

200   RETURN

      END
```

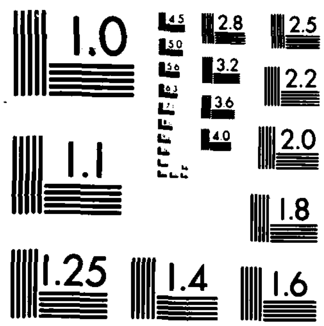
REDUCE.F

```
      SUBROUTINE ZZREDC(AA,B,LINEAR)

C     *** REDUCES DIMENSIONAL JACOBIAN TO NON DIMENSIONAL DERIVATIVES ***

      PARAMETER (ISIZE=5,ISIZEM1=4)          !*** SIZE OF THE PROBLEM ***
      LOGICAL   FLAG, LINEAR
      CHARACTER*1Ø C(11)
      DIMENSION AA(ISIZE,ISIZE),B(24),D(11)
      REAL IYYND, LT

      DATA C /'CTV      =','CLV      =','CDV      =','CMV      =',
     1        'CLALPHA  =','CDALPHA  =','CMALPHA  =','CLDALPHA =',
     1        'CMDALPHA =','CLQ      =','CMQ      ='/

C     *** WRITE NON-DIMENSIONAL JACOBIAN TO FILE ***

      WRITE (9,5Ø)

1Ø    DO 2Ø I=1,ISIZEM1

      WRITE (9,7Ø) (AA(I,J),J=1,ISIZEM1)

2Ø    CONTINUE

C     *** RECONSTITUTE FLIGHT DATA ***

         SGA   =B(1)
         CGA   =B(2)
         SAL   =B(3)
         CAL   =B(4)
         CWE   =B(5)
         CLE   =B(6)
         CTE   =B(7)
         CDE   =B(8)
         AMU   =B(9)
         IYYND =B(1Ø)
         XLT   =B(11)
         XEPSAL=B(12)
         CW    =B(13)

      IF ( LINEAR ) THEN

C     *** APPROXIMATE SURPLUS DERIVATIVE  CTV ***

         D(1)  = -3.Ø*CTE                                  !CTV

C     *** CALCULATE NON-DIMENSIONAL DERIVATIVES ***

         D(6)  = CLE-2.Ø*AMU*AA(1,2)                       !CDALPHA
         D(3)  = D(1)*CAL+2.Ø*CWE*SGA-2.Ø*AMU*AA(1,1)      !CDV
         D(11) = IYYND*AA(3,3)/(1.Ø+XEPSAL*AA(2,3))        !CMQ
         D(9)  = D(11)*XEPSA'                              !CMDALPHR
         D(1Ø) = -D(11)*CW/XLT                             !CLQ
         D(8)  = XEPSAL*D(1Ø)                              !CLDALPHA

         DENOM = 2.Ø*AMU+D(8)                              !DENOMINATOR
```

```
         D(5)   = -AA(2,2)*DENOM-CDE                          !CLALPHA
         D(2)   = -D(1)*SAL-AA(2,1)*DENOM-2.0*CWE*CGA         !CLV
         D(4)   = IYYND*AA(3,1)-D(9)*AA(2,1)                  !CMV
         D(7)   = IYYND*AA(3,2)-D(9)*AA(2,2)                  !CMALPHA

      ELSE

C     *** APPROXIMATE SURPLUS DERIVATIVE  CTV ***

         D(1)  =-3.*CTE                                       ! CTV

C     *** CALCULATE NON-DIMENSIONAL DERIVATIVES ***

         D(8)  = -CWE*SGA/AA(2,4)-2.*AMU                      ! CLDALPHA

         DENOM= 2.0*AMU+D(8)                                  ! DENOMINATOR

         D(9)  = -AA(3,4)*IYYND*DENOM/CWE/SGA                 ! CMDALPHA
         D(6)  = CLE-2.*AMU*AA(1,2)                           ! CDALPHA
         D(5)  = -AA(2,2)*DENOM-CDE                           ! CLALPHA
         D(10)= 2.*AMU-AA(2,3)*DENOM                          ! CLQ
         D(3)  = D(1)*CAL+2.*CWE*SGA-2.*AMU*AA(1,1)           ! CDV
         D(2)  = -D(1)*SAL-2.*CWE*CGA-DENOM*AA(2,1)           ! CLV

         FACT1= D(9)*(D(1)*SAL+D(2)+2.*CWE*CGA)/DENOM

         D(4)  = IYYND*AA(3,1)+FACT1                          ! CMV
         D(7)  = IYYND*AA(3,2)+D(9)*(D(5)+CDE)/DENOM          ! CMALPHA
         D(11)= IYYND*AA(3,3)-D(9)*(2.*AMU-D(10))/DENOM! CMQ

      ENDIF

C     *** OUTPUT COEFFICIENTS TO FILE SDOFAP.L ***

      WRITE(9,60)
      DO 30 II=1,11
      WRITE(9,40) C(II),D(II)
30    CONTINUE
      WRITE(9,80)

40    FORMAT(/,4X,A10,2X,F14.7)
50    FORMAT(////,35H  *** NON-DIMENSIONAL JACOBIAN. ***,//)
60    FORMAT(////
     .,50H  *** NON-DIMENSIONAL AERODYNAMIC DERIVATIVES. ***,//)
70    FORMAT(/,4X,F11.7,2X,F11.7,2X,F11.7,2X,F11.7)
80    FORMAT(///)

      RETURN

      END


      SUBROUTINE ACINFO(CTP,SWING,CWING,ALPT,XLT,XEPSAL)


      INCLUDE 'FTPAR.F'

         CTP=CTPROP
         SWING=SW
         CWING=CW
         ALPT=THSET
```

```
C     *** CALCULATE THE TAILPLANE'S MOMENT ARM ( XLT ) ***

         XLT = XT - XQARTC

C     *** THE RATE OF CHANGE OF DOWNWASH WITH ALPHA IS PREFIXED WITH ***
C     *** AN 'X' TO OVERCOME PROBLEMS WITH THE COMMON STATEMENT.     ***

         XEPSAL = EPSAL

      RETURN

      END
```

AD-A173 849   A FLIGHT DYNAMIC SIMULATION PROGRAM IN AIR-PATH AXES   2/2
USING ACSL (ADVANCED..(U) AERONAUTICAL RESEARCH LABS
MELBOURNE (AUSTRALIA) P W GIBBENS JUN 86
UNCLASSIFIED   ARL-AERO-TM-380   F/G 12/1   NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

DISTRIBUTION

AUSTRALIA

<u>Department of Defence</u>

<u>Defence Central</u>

Chief Defence Scientist)
Deputy Chief Defence Scientist)
Superintendent, Science and Program Administration) 1 Copy
Controller, External Relations, Projects and
Analytical Studies)
Defence Science Counsellor (UK) (Doc Data Sheet Only)
Counsellor, Defence Science (USA) (Doc Data Sheet Only)
Defence Science Representative (Bangkok)
Defence Central Library
Document Exchange Centre, DISB (18 copies)
Joint Intelligence Organisation
Librarian H Block, Victoria Barracks, Melbourne

<u>Aeronautical Research Laboratories</u>

Director
Library
Superintendent - Aerodynamics
Divisional File - Aerodynamics
D. Secomb
R. Feik
C. Martin
T. Tran-Cong
J. Drobik
M. Cooper
Author: P.W. Gibbens

<u>Materials Research Laboratories</u>

Director/Library

<u>Defence Research Centre</u>

Library

<u>Navy Office</u>

Navy Scientific Adviser
RAN Aircraft Maintenance and Flight Trials Unit
Directorate of Naval Aircraft Engineering

<u>Army Office</u>

Scientific Adviser - Army
Royal Military College Library

Air Force Office
Air Force Scientific Adviser
Aircraft Research and Development Unit
Scientific Flight Group
Library
Technical Divisional Library
Director General Aircraft Engineering - Air Force
Director General Operational Requirements - Air Force
HQ Support Command (SLENGO)
RAAF Academy, Point Cook

Government Aircraft Factories
Manager
Library

Department of Aviation
Library
Flight Standards Division

Industry
Commonwealth Aircraft Corporation, Library
Hawker de Havilland Aust. Pty. Ltd, Bankstown, Library

Universities and Colleges

Adelaide
Barr Smith Library

Flinders
Library

La Trobe
Library

Melbourne
Engineering Library

Monash
Hargrave Library

Newcastle
Library

Sydney
Engineering Library
Prof G.A. Bird, Dept. of Aeronautical Engineering

NSW
Physical Sciences Library

Queensland
Library

Tasmania
    Engineering Library

Western Australia
    Library

RMIT
    Library

SPARES (15 copies)

TOTAL (83 copies)

Department of Defence

DOCUMENT CONTROL DATA

1.a. AR No	1.b. Establishment No	2. Document Date	3. Task No
AR-004-472	ARL-AERO-TM-380	JUNE 1986	DST 82/030

4. Title	5. Security	6. No Pages	7. No Refs
A FLIGHT DYNAMIC SIMULATION PROGRAM IN AIR-PATH AXES USING ACSL	a. document: UNCLASS. b. title: U c. abstract: U	90	6

8. Author(s)	9. Downgrading Instructions
P. W. GIBBENS	

10. Corporate Author and Address	11. Authority (as appropriate) a.Sponsor b.Security c.Downgrading d.Approval
Aeronautical Research Laboratories P.O. Box 4331, MELBOURNE, VIC. 3001	

12. Secondary Distribution (of this document)

Approved for public release.

Overseas enquirers outside stated limitations should be referred through ASDIS, Defence Information Services Branch, Department of Defence, Campbell Park, CANBERRA ACT 2601

13.a. This document may be ANNOUNCED in catalogues and awareness services available to ...

No limitations.

13.b. Citation for other purposes (ie casual announcement) may be (select) unrestricted(or) as for 13 a.

14. Descriptions	15. COSATI Group
Equations of motion Flight simulation Flight dynamics Advanced Continuous Simulation Language Air path axes	01010

16. Abstract

The six degrees of freedom dynamic equations of motion have been programmed in the Advanced Continuous Simulation Language (ACSL) for use in aircraft simulations at ARL. Air-path axes were chosen for the integration of the force equations, and body axes for the integration of the moment equations. The use of quaternions for the determination of the direction cosines has been described.

This paper is to be used to record information which is required by the Establishment for its own use but which will not be added to the DISTIS data base unless specifically requested.

16. Abstract (contd)

17. Imprint

Aeronautical Research Laboratories, Melbourne

18. Document Series and Number	19. Cost Code	20. Type of Report and Period Covered
Aerodynamics Technical *Memorandum* 380	52 7710	Written September 1985.

21. Computer Programs Used

22. Establishment File Ref(s)

END

12-86

DTIC